commodore 16

COMMODORE 16
Guida all'Utente

INDICE

CAPITOLO 1

INSTALLAZIONE

- Cosa c'è in questa scatola
- Impariamo a conoscere gli interruttori e i connettori
- Installazione del Commodore 16
- Tavola dei problemi
- Periferiche

INSTALLAZIONE DEL COMMODORE 16 Cosa c'è in questa scatola?

Avete aperto la scatola che contiene il vostro nuovo Commodore 16 e avete trovato questo manuale. La prima cosa che dovreste fare è controllare che tutti gli oggetti qui elencati ci siano. Dovreste avere:

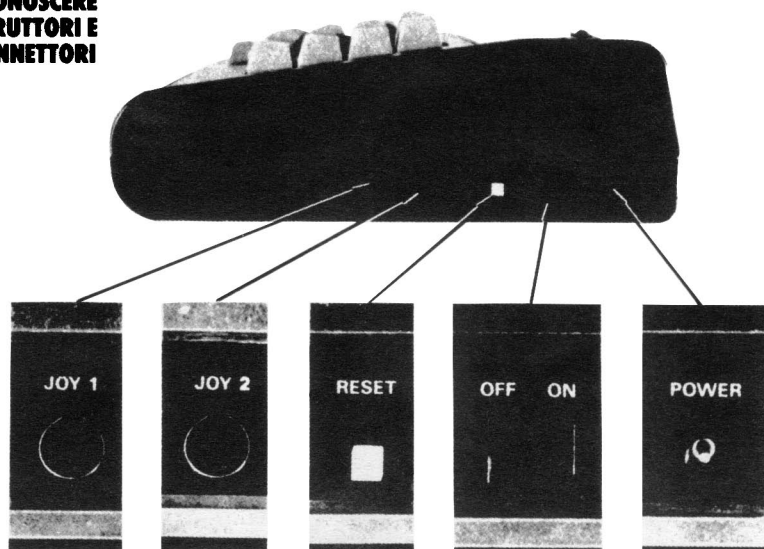
1 Il Commodore 16

2 L'alimentatore

3 Il cavo RF

4 Il manuale per l'utente (probabilmente lo avete già trovato, dato che lo state leggendo!)

Se manca qualcosa fermi tutti! Rivolgetevi immediatamente al vostro rivenditore per avere un'altra confezione.
Prima di collegare questi componenti fra di loro dovreste guardare attentamente queste foto familiarizzandovi con le prese e gli interruttori, potrete installare il vostro sistema facilmente e velocemente.

IMPARIAMO A CONOSCERE INTERRUTTORI E CONNETTORI

Lato Destro del Commodore 16

L'interruttore di accensione

Il vostro Commodore 16 deve essere spento (OFF) quando inserite o togliete le CARTRIDGES o qualunque periferica come la STAMPANTE o il DATASETTE. C'è una spia rossa di accensione a destra sopra la tastiera, che si illumina quando il computer è acceso.

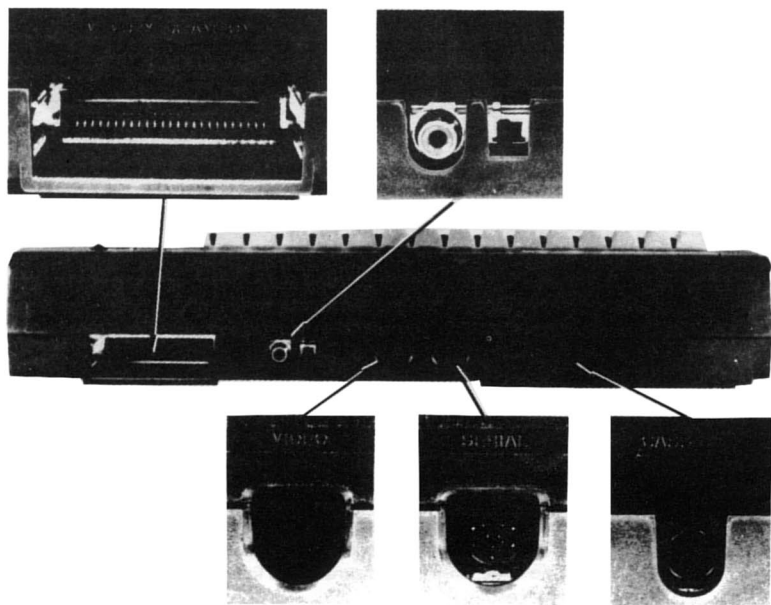
La presa dei joystick

Potete inserire il joystick in queste prese, chiamate JOY 1 e JOY 2. Il Commodore 16 usa dei joystick speciali disponibili presso il vostro rivenditore Commodore.

Pulsante di reset

Premete il pulsante di reset quando volete ricominciare da capo al punto in cui eravate quando avevate appena acceso il computer. Il pulsante di reset cancella il video e cancella qualsiasi programma BASIC precedentemente digitato. Cancellare il video senza cancellare i programmi è possibile, ma verrà spiegato nel capitolo 4.

La presa di alimentazione

Un'estremità del cavo di alimentazione si inserisce qui. L'altra va inserita in una normale presa di corrente.

Il retro del computer

La porta per le espansioni di memoria

Le cartridges contenenti il software per il Commodore 16 si inseriscono qui. Prima di inserire o estrarre le cartridges controllate che il computer sia spento (OFF).

L'uscita TV

Solo una delle due estremità del cavo TV (quello nero sottile) deve essere inserita qui. L'altra invece deve essere inserita nella presa apposta del vostro televisore.

L'uscita video

Il cavo di collegamento fra il monitor e il Commodore 16 deve essere inserito qui. Se collegate il computer ad un televisore, questo non sarà necessario.

L'uscita seriale

A questa presa può essere collegata l'unità a disco o la stampante. Volendo collegarle insieme, collegate prima l'unità a disco a questa presa, poi, collegate lo spinotto della stampante sul retro dell'unità a disco.

L'uscita per l'unità a cassette

Va collegato a questa uscita il registratore Commodore 1531 Datassette per cassette contenenti software.

INSTALLAZIONE DEL COMMODORE 16

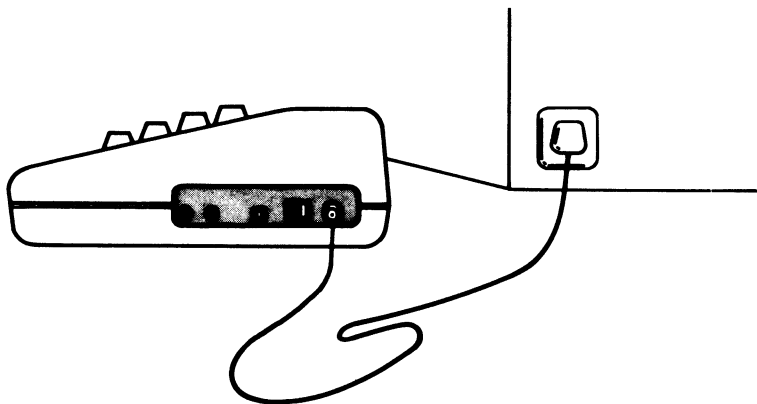
Collegare il vostro Commodore 16 è facile come bere un bicchiere d'acqua! Dovete fare solo tre cose:

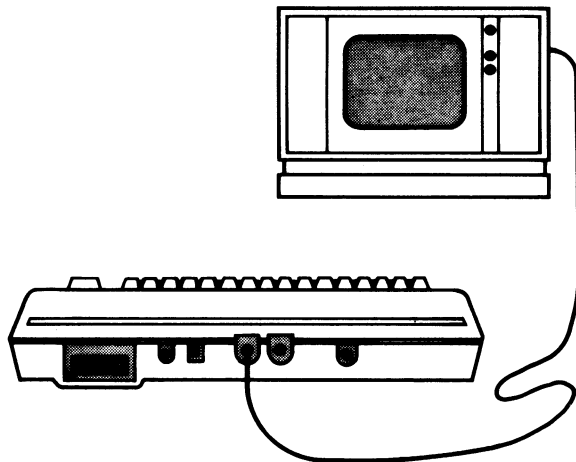
1 Collegare l'alimentatore al lato destro del computer e ad una presa di corrente.

2 Collegate il cavo RF (quello sottile nero) dal commutatore d'antenna all'uscita RF del computer.

Assicuratevi di avere un numero sufficiente di prese di corrente per connettere tutte le spine alla presa di corrente più vicina. Forse avrete bisogno di una spina multipla o di una prolunga, specialmente per collegare anche la stampante e l'unità a disco. Ricordatevi che tutti gli apparecchi (computer, TV, etc) devono essere spenti (OFF) fino a quando tutto è stato connesso ed è pronto per funzionare.

1 Collegamento dell'Alimentatore

Prima inserite lo spinotto rotondo dell'alimentatore nella presa di alimentazione (POWER) sul lato destro del computer, poi connettete l'altra spina alla presa di corrente di rete.

2 Collegamento del Cavo TV

Inserite una estremità del cavo TV nella presa del vostro televisore.
Inserite l'altra estremità del cavo nella presa TV del computer.

Collegamenti fra il Commodore 16 e il Monitor

Se avete intenzione di connettere il vostro Commodore 16 ad un monitor invece che ad un televisore, seguite le istruzioni allegate al monitor. Collegarsi ad un monitor come il Commodore Color Monitor TV è facile. Infatti è sufficiente connettere un solo cavo che va direttamente dal monitor alla Presa VIDEO sul retro del computer.

Per finire...

Ora è giunto il momento di accendere il computer. (Se siete stati attenti dovreste sapere dove si trova l'interruttore di accensione).

Se tutto è andato bene la spia rossa di accensione si illumina, e sullo schermo appare questo messaggio

```
COMMODORE BASIC 3.5 12277 BYTES FREE
   READY
```

Il cursore lampeggiante sotto il messaggio READY vi informa che il vostro Commodore 16 è in attesa di comandi. Il colore di sfondo è bianco, mentre le lettere sono stampate in nero. Il bordo dello schermo è viola.

Se qualcosa non funziona vi sarà utile la TAVOLA CERCAPROBLEMI.

TAVOLA CERCAPROBLEMI

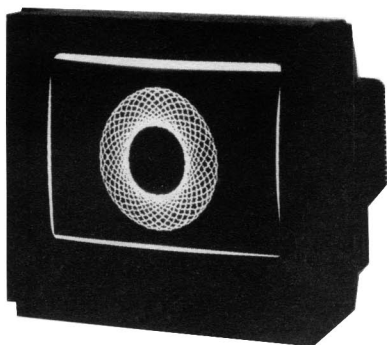
Sintomo	Causa	Rimedio
Spia di accensione spenta	Computer spento Cavo di alimentazione non inserito Alimentatore non connesso Fusibile rotto	Assicurarsi che l'interruttore sia su ON Controllare la connessione con la presa Controllare le due prese di corrente Portare il sistema da un rivenditore per sostituire il fusibile
Nessuna immagine	TV sintonizzata sul canale sbagliato Cavo TV sconnesso	Cercare il canale giusto Controllare le connessioni del cavo RF
Disegni casuali sul video	Cartridge inserita in modo errato	Reinserire la cartridge dopo aver spento il computer
Immagine senza colore	TV sintonizzata male TV non ben collegata Controllo colore troppo basso sulla TV o sul computer	Risintonizzare il TV Controllare le connessioni Sistemare il controllo di colore
Immagine OK ma niente audio	Volume del televisore troppo basso Televisore mal sintonizzato	Sistemare il volume Risintonizzare il TV

IMPORTANTE: Alcuni televisori non riescono a mostrare l'intero schermo dell'immagine. Si consiglia dunque di usare un altro televisore o un monitor come i Commodore 1701, 1702, 1802 o 1803 a colori.
Se questo non è possibile, potete evitare il problema premendo il tasto "ESC" seguito dal tasto "R". Questo riduce la dimensione dello schermo ad una dimensione di 38 colonne, cosicché l'intera immagine può stare nello schermo. Dovrete ripetere questo ogni volta che accendete o resettate il computer. Il software protetto da terzi non permette però questa opinione.

PERIFERICHE

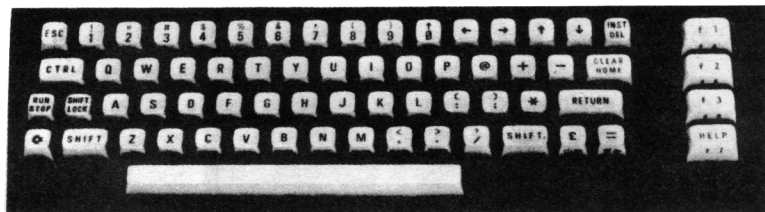
Le periferiche sono accessori che potete aggiungere al Commodore 16 per aumentarne le applicazioni. Sono disponibili presso i rivenditori Commodore, e vi permettono di sfruttare al massimo le possibilità del vostro computer. Con le periferiche potete salvare e immagazzinare dati, stampare ciò che appare sullo schermo (in bianco e nero o a colori), usare programmi memorizzati su cassetta o dischetto e così via.

Per salvare o caricare programmi, avrete bisogno di una periferica che immagazzina dati. I dati possono essere salvati e caricati sia da cassette che da dischetti. Per usare software su cassetta (e per salvare i vostri programmi su cassetta) avrete bisogno del registratore Commodore 1531 DATASSETTE. Per i dischetti si possono usare varie unità a disco. Le unità a disco sono particolarmente veloci ed efficienti. I modelli Commodore 1541 e 1551 sono appunto unità a disco compatibili con il Commodore 16.

Non sempre i normali televisori danno immagini perfettamente nitide. I monitor a colori della Commodore sono progettati per darvi le immagini più chiare e luminose. Ci sono vari modelli disponibili, tra cui i Commodore 1701, 1702, 1802 e 1803.

Ci sono varie stampanti Commodore che possono funzionare con il vostro computer. Tra queste la 1520 Printer/plotter (che può stampare sia grafici che testi), la MCS-801, MPS-802, MPS-803 (con il tractor-feed opzionale) e la DPS-1101 (alta definizione)

CAPITOLO 2

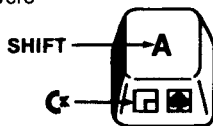
LA TASTIERA

UNA PANORAMICA DELLA TASTIERA

La maggior parte dei tasti che vedete sul Commodore 16 sono identici a quelli di una macchina da scrivere. Ogni tasto poi può fare molto di più, esistono tasti speciali come il tasto Commodore C= e i quattro tasti di cursore separati. Vi mostreremo le possibilità addizionali di ogni tasto, e come usare i simboli grafici che compaiono su molti tasti. Poi voi dovrete trovare i tasti ed allenarvi ad usarli.

Usare la Tastiera come quella di una Macchina da Scrivere

Quando voi digitate per la prima volta delle lettere sul vostro computer, esse appaiono sullo schermo come maiuscole. Le lettere e i numeri appaiono sullo schermo esattamente come sono sui tasti che premete. Anche altri tasti +, –, =, @, *, e £ (il simbolo della sterlina inglese) possono essere premuti da soli. La maggior parte della punteggiatura può essere raggiunta solo con il tasto SHIFT. Se volete scrivere in modo "normale", potete scrivere sia in maiuscolo che in minuscolo (come in una macchina da scrivere) premendo contemporaneamente il tasto SHIFT e il tasto C=. Dopo aver fatto ciò, tutte le lettere battute da sole verranno stampate in minuscolo. Quando premerete lo SHIFT insieme ad una lettera, avrete una lettera maiuscola. I numeri e la punteggiatura funzioneranno in modo normale (maiuscole e caratteri grafici). Per uscire basterà premere ancora i due tasti (SHIFT e C=) insieme.

Modo macchina da scrivere

TASTO	+		TASTO =
Shift	+	C=	= "macchina da scrivere"
A	+		= a
Shift	+	A	= A
C=	+	A	= ┌

Tasti Speciali

Molti tasti del vostro Commodore 16 si comportano in modo assai differente dalle altre macchine da scrivere. Servono a far si che altri tasti facciano cose che normalmente non fanno, o per fargli svolgere funzioni particolari. La tastiera del Commodore 16 contiene anche dei simboli speciali che non si trovano su molte macchine da scrivere o addirittura su molti computers. Questi simboli speciali comprendono il simbolo della sterlina inglese (£), pi (π), i simboli di minore e maggiore (<>), le parentesi ([]), e le frecce (↑←). Nei programmi del Commodore 16, questi tasti "speciali" sono usati spesso.

RETURN

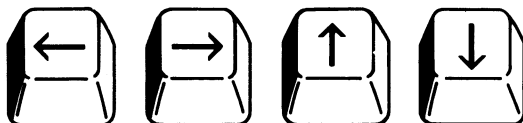
Il tasto RETURN va premuto alla fine di ogni linea di istruzioni digitata sulla tastiera del Commodore 16. RETURN serve a far entrare informazioni e istruzioni nel vostro computer.

SHIFT

Avete già incontrato il tasto SHIFT quando avete usato la tastiera come una macchina da scrivere. Questa è la tipica funzione del tasto SHIFT: esso è sempre usato per modificare ciò che gli altri tasti scrivono sullo schermo, ma da solo non serve a niente. Il tasto SHIFT vi dà la possibilità di scrivere lettere maiuscole, simboli grafici, la punteggiatura ed altre cose, con un po' di aiuto da parte di altri tasti. Potrete vedere altre funzioni dello SHIFT in questa sezione, come per esempio per raggiungere i simboli grafici.
Il tasto SHIFT LOCK funziona come lo SHIFT, eccetto per il fatto che resta premuto, cosicché non è necessario tenerlo schiacciato. Quando SHIFT LOCK è premuto, tutti i caratteri sono stampati come se il tasto SHIFT fosse sempre schiacciato. Per rilasciare il tasto basta premerlo una seconda volta, e tutto sarà stampato in modo normale.

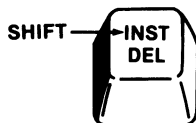
RUN/STOP

Premendo questo tasto mentre un programma stà "girando", il Commodore 16 interromperà ciò che sta facendo e il controllo viene ridato alla tastiera. Premendo contemporaneamente i tasti SHIFT e RUN/STOP, il Commodore 16 caricherà il primo programma sul disco dall'unità a disco se si trova in "modo Programma").

I TASTI DI CONTROLLO DEL CURSORE

Il cursore è un quadratino lampeggiate che indica dove vi trovate sullo schermo del computer e può essere spostato velocemente e facilmente per tutto lo schermo usando i TASTI DI CONTROLLO DEL CURSORE. Ci sono quattro tasti separati per il cursore, ognuno con una freccia che ne indica la direzione di spostamento: su, giù, sinistra, o destra. Potete usare i tasti di controllo del cursore per spostarlo su tutto lo schermo senza che i caratteri vengano modificati. Come tutti i tasti del Commodore 16, ciascun tasto di controllo del cursore ha il 'repeat' automatico. Questo significa che se tenete premuto il tasto in continuazione, il cursore continua a muoversi nella direzione del tasto che avete premuto (questo vi risparmia la noia di ticchettare furiosamente il tasto per ottenere lunghi spostamenti).

INST/DEL

Voi potete inserire (INSERT) e cancellare (DELETE) lettere e numeri dalla linea che state scrivendo premendo questo tasto. Quando premete questo tasto da solo, il carattere subito a sinistra del cursore scompare, e il cursore si sposta dove prima c'era quel carattere. Potete usare i tasti di controllo del cursore per andare in mezzo ad una riga e usare DEL per cancellare una lettera. Quando fate ciò, la lettera a sinistra viene cancellata, mentre le lettere nel resto della linea vengono spostate a sinistra di un carattere per coprire il "buco".
Per creare spazio premete questo tasto insieme allo SHIFT. Lo spazio viene creato alla destra del cursore; mentre questo resta fermo. Quando inserite uno spazio nel mezzo di una linea, il resto della linea viene spostato a destra.
Il tasto INS/DEL vi farà risparmiare molto tempo quando dovrete mettere a posto o cambiare testo.

TASTO	+ TASTO	EFFETTO
INST/DEL	+	= Delete (cancella i caratteri digitati)
SHIFT	+ INST/DEL	= Insert (inserisce i caratteri)

Clear Home

Questo tasto possiede due funzioni: HOME e CLEAR. Quando lo premete, il cursore si sposta immediatamente nell'angolo in alto a sinistra dello schermo (che è conosciuto come posizione di HOME). Il resto dello schermo rimane uguale. Se tenete premuto il tasto SHIFT e premete CLEAR HOME, non solo il cursore si muove nella posizione di HOME, ma tutto lo schermo viene cancellato. Tutto ciò che rimane sullo schermo è il cursore lampeggiante nell'angolo in alto a sinistra dello schermo.

TASTO	+ TASTO	EFFETTO
Clear Home	+	= Posizione di home
Clear Home	+ Shift	= cancella schermo

CTRL (control)

Il tasto CTRL come il tasto SHIFT, funziona solo se è premuto insieme ad altri tasti. È usato in tre casi:

1) Come è spiegato nella sezione TASTI PER IL COLORE, premendo CTRL e un tasto di controllo colore potete cambiare colore al testo stampato sullo schermo.
2) Potete fermare temporaneamente un programma che stampa o un listato sullo schermo premendo CTRL e il tasto S. (Per far continuare il programma premere un tasto qualunque).
3) CTRL è anche usato con REVERSE ON/OFF e FLASH ON/OFF tasti di cui parleremo poi.

In più, alcuni programmi fanno uso del tasto control per funzioni speciali.

C= (tasto Commodore)

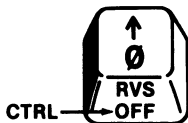
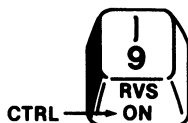
Il tasto Commodore è molto simile al CTRL e può essere usato per ottenere queste funzioni:

1) Quando è usato col tasto SHIFT, il tasto C= seleziona il "modo" macchina da scrivere, nel quale potete usare lettere sia maiuscole che minuscole
2) Il tasto C= funziona sempre come uno shift per permettervi di stampare il simbolo grafico posto a sinistra su di ogni tasto. Basta che teniate premuto C= mentre selezionate il simbolo grafico voluto.
3) Il tasto C= è usato come il CTRL per cambiare il colore di ciò che stampate sullo schermo se premuto con un tasto di CONTROLLO COLORE.

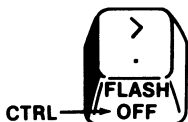
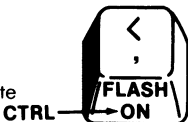
4) Quando volete rallentare uno scrolling (scorrimento) sullo schermo (per esempio un listato che appare linea per linea sullo schermo troppo veloce per essere seguito), tenete premuto il tasto C= . La velocità di scorrimento dello schermo diminuisce considerevolmente. Quando rilasciate il tasto, ritorna a velocità normale. (Ebbene sì! Il tasto C= può fare qualche cosa da solo!)

RVS ON RVS OFF (Reverse On/Off)

Il Commodore 16 vi permette di stampare lettere e numeri in negativo (reverse). In altre parole, se il cursore è nero e lo sfondo è giallo, ciò che digitate appare con lettere gialle su uno sfondo nero.
Ecco tutto ciò che dovete fare per ottenere la stampa in negativo: premete il tasto CTRL e il tasto RVS ON. Tutto ciò che digiterete d'ora in avanti apparirà in negativo, fino a quando premerete CTRL e RVS OFF, o il tasto RETURN, o il tasto ESC e O. Questo vi permette di ritornare in modo normale (non in negativo).

TASTO	+ TASTO	= EFFETTO
CTRL	+ RVS ON	= Stampa in negativo
CTRL	+ RVS OFF	= Stampa Normale

FLASH ON FLASH OFF

Potete fare lampeggiare i caratteri sullo schermo continuamente, così come fa il cursore. Premete i tasti CTRL e FLASH ON per fare lampeggiare tutto ciò che digitate. Premendo CTRL e FLASH OFF, o RETURN, o ESC tornerete al modo normale (non lampeggiante).

TASTO	+ TASTO	= EFFETTO
CTRL	+ FLASH ON	= Stampa lampeggiante
CTRL	+ FLASH OFF	= Stampa normale

ESC

Il tasto ESC vi permette di usare molte funzioni speciali per l'editing di schermo, comprese le funzioni per costruire e manipolare le finestre di schermo. Le finestre sono una speciale caratteristica del Commodore 16, che vi permettono di riservare un'area che può essere usata come spazio di lavoro senza influenzare il resto dello schermo. Il tasto ESC può compiere numerose funzioni riguardanti le finestre e non, come inserimenti, cancellazioni, e scorrimenti. Tutte le funzioni e il tasto ESC saranno elencate e spiegate nel Capitolo 4.

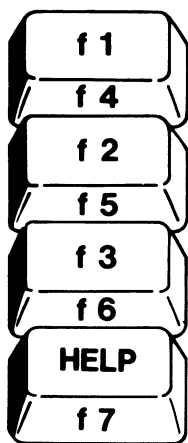
Tasti di Controllo Colore

I tasti di controllo colore corrispondono ai tasti numerici da 1 a 8. Quando sono premuti assieme al tasto CTRL o C= cambiano il colore di tutto ciò che digitate. Il Commodore 16 si presenta all'accensione con uno sfondo bianco, un bordo viola e un cursore lampeggiante nero. Ciò che scrivete appare in nero. Per cambiare il colore del cursore (e di ciò che scrivete), usate i tasti di controllo colore. Sulla fronte dei tasti numerati da 1 a 8, sono scritti due colori. Premete CTRL con un tasto numerico per ottenere il colore scritto sopra, C= con un tasto numerico per ottenere quello scritto sotto. Potete cambiare solo il colore dei caratteri usando i tasti di controllo colore. Per cambiare il colore dello sfondo o del bordo, dovete usare un comando BASIC, che sarà spiegato in seguito.

Tasto	= Effetto	Tasto	= Effetto
CTRL + 1	= Nero	C= + 1	= Arancio
CTRL + 2	= Bianco	C= + 2	= Marrone
CTRL + 3	= Rosso	C= + 3	= Giallo Verde
CTRL + 4	= Ciano	C= + 4	= Rosa
CTRL + 5	= Viola	C= + 5	= Blu Verdino
CTRL + 6	= Verde	C= + 6	= Blu Chiaro
CTRL + 7	= Blu	C= + 7	= Blu Scuro
CTRL + 8	= Giallo	C= + 8	= Verde Chiaro

Tasti Grafici

Ogni tasto alfabetico (e anche alcuni altri) ha due quadratini sulla fronte del tasto, ognuno con una differente linea o simbolo. Questi sono i tasti grafici. Appena acceso il vostro Commodore 16 stampa con lettere maiuscole; se tenete premuto il tasto SHIFT potrete stampare il simbolo grafico mostrato sulla fronte del tasto. Potete stampare l'intero set di più di 60 caratteri grafici che vedete sulla fronte dei tasti. Potete creare immagini, grafici e disegni stampando questi caratteri fianco a fianco o uno sopra l'altro, come mattonelle, o rendere più interessanti i vostri grafici usando i tasti di colore.
Provate a stampare qualche simbolo grafico per vedere come funzionano. Il Capitolo 5 dà ulteriori spiegazioni sui grafici.
Quando siete nel modo macchina da scrivere, potete usare solo i simboli grafici posti sulla sinistra (premendo C= e il tasto appropriato).
I simboli grafici sulla sinistra sono l'ideale per disegni, grafici e tabelle.

Tasti di Funzione

I quattro tasti di funzione sulla parte destra della tastiera (staccati dagli altri) sono speciali tasti di funzione che vi permettono di risparmiare tempo eseguendo funzioni ripetitive premendo un solo tasto. Portano le scritte f1, f2, f3 e HELP. Potete avere queste funzioni semplicemente premendo uno di questi tasti. La fronte dei tasti porta le scritte f4, f5, f6 e f7. Premete SHIFT e f1, f2, f3 e HELP rispettivamente, per ottenere queste funzioni.

Ecco cosa fà ogni tasto:

TASTO 1 seleziona uno dei modi grafici (GRAPHICS) dopo che avete selezionato il modo e avete premuto RETURN. Il comando GRAPHICS è necessario per dare comandi grafici come CIRCLE o PAINT. Per ulteriori informazioni vedere il capitolo 6.
TASTO 2 stampa 'DLOAD" sullo schermo. Tutto ciò che dovete fare è di inserire il nome del programma da caricare da disco e premere RETURN.
TASTO 3 lista un DIRECTORY (elenco) dei files presenti sul disco.
TASTO 4 cancella lo schermo.
TASTO 5 stampa 'DSAVE" sullo schermo. Tutto ciò che dovete fare è di inserire il nome del corrente programma da salvare sul disco e premere RETURN.
TASTO 6 lancia (RUN) il programma in memoria.
TASTO 7 visualizza un listato del programma.
TASTO 8 (il tasto HELP) evidenzia gli errori nelle linee di programma facendoli lampeggiare.

Con alcune semplici istruzioni Basic i tasti di funzione possono essere ridefiniti. Si tratta di un opzione utile quando si ha spesso a che fare con una complicata serie di istruzioni. Basterà schiacciare un tasto chiave precedentemente ridefinito per ottenerla.
Una volta spento il computer i tasti tornano alla loro normale funzione.

Il Tasto Help (aiuto)

Quando fate un errore in un programma, il computer visualizza un messaggio di errore.
Dopo un messaggio di errore, premete il tasto HELP per localizzarlo esattamente. Quando premete HELP, la linea sbagliata è visualizzata sullo schermo con l'errore lampeggiante. Per esempio:

? SINTAX ERROR IN LINE 10	Il computer visualizza questo
HELP	Premete HELP
10 PRONT "COMPUTER COMMODORE"	L'errore è visualizzato facendolo lampeggiare

CAPITOLO 3

SOFTWARE

- Introduzione
- Cartridges
- Cassette
- Dischetti

INTRODUZIONE

"Software" indica tutti i programmi che possono essere immessi ed eseguiti in un computer. L'hardware (in questo caso il vostro Commodore 16) può usare il software in varie forme. Cartridges (cartucce) da inserire nel computer, cassette e dischetti pre-registrati. La quantità di software disponibile per il Commodore 16 stà aumentando velocemente.
Il Commodore 16 può usare software su CARTRIDGE, CASSETTE MAGNETICHE, e DISCHETTI, disponibili presso il vostro rivenditore Commodore. Tutto ciò che dovete fare è di caricarli nel Commodore 16. Potete anche creare e immagazzinare i vostri Programmi usando nastri magnetici o floppy disks.

CARTRIDGES (cartucce)

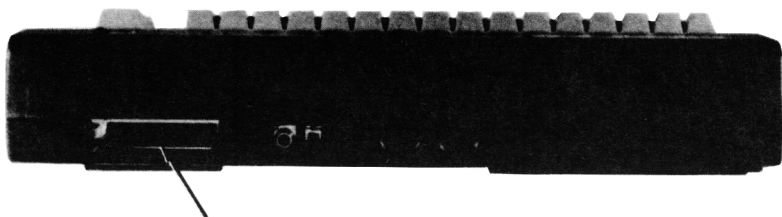
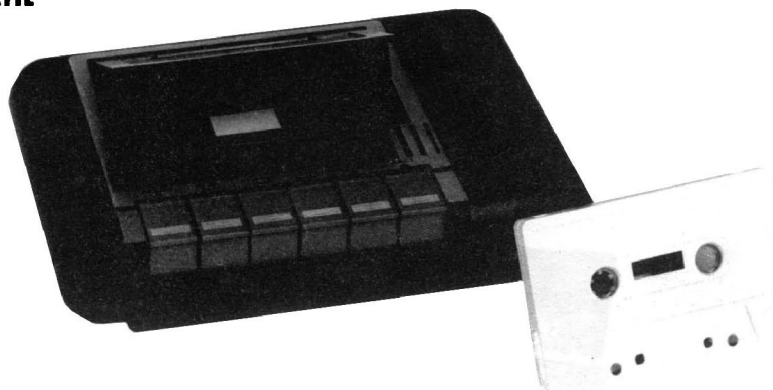
La Commodore produce un vasto assortimento di software su cartridge per il vostro Commodore 16, una varietà di programmi personali, educativi, e professionali, così come eccitanti giochi disponibili già da adesso. Per far funzionare le cartridges non avete bisogno di una apparecchiatura addizionale. Tutto ciò che dovete fare è inserire la cartridge nel retro del computer ed accenderlo. Ecco i vari passi da seguire per usare le cartridges:

1 SPEGNERE il Commodore 16.

IMPORTANTE: dovete spegnere il computer prima di inserire o rimuovere le cartridges. Se non lo fate potreste danneggiare sia le cartridges che il computer.

2 Tenete la cartridge con l'etichetta verso l'alto e inseritela a fondo nella fessura apposita (etichettata "memory ex pansion") nel retro del computer.

3 ACCENDERE il Commodore 16.

4 Iniziare il gioco o il programma a seconda delle istruzioni fornite con il programma.

CASSETTE

Una grande varietà di software è disponibile su cassette magnetiche simili alla musicassette che ascoltate su normali registratori. Il software su cassetta funziona allo stesso modo del cartridge, ma per caricare il software è necessario un apparecchio particolare (chiamato periferica): un registratore Datassette, disponibile presso il vostro rivenditore Commodore.
Potete anche usare le cassette e il Datassette per memorizzare i programmi scritti da voi stessi. I passi da seguire per caricare un programma da cassetta sono gli stessi sia che stiate usando un nastro pre-registrato o un programma che avete salvato voi stessi.

Come Caricare le Cassette

1 Inserite la cassetta nel Datassette e chiudete lo sportellino.

2 Riavvolgete la cassetta premendo il tasto REWIND sul Datassette.

3 Quando il nastro è completamente riavvolto schiacciate il pulsante STOP sul DATASSETTE digitate LOAD e premete RETURN Il computer risponde con il messaggio:
PRESS PLAY ON TAPE

4 Premete il pulsante PLAY sul Datassette. Lo schermo si spegne e il Datassette si avvia. Quando viene incontrato un programma, appre questo messaggio:
SEARCHING
FOUND nome del programma

5 Premete il tasto Commodore per caricare il programma appena trovato (FOUND). Se c'è più di un programma sul nastro, e il programma che il Commodore 16 ha trovato non è quello che volevate non fate nulla. Dopo una breve pausa riprenderà la ricerca.

Quando un programma è stato caricato, appare la parola READY. Se volete fermare il caricamento prima che questo sia completato premete il tasto RUN/STOP sulla tastiera, e poi il tasto STOP sul Datassette. Dopo che il software è stato caricato, digitate RUN per far partire il programma. Potete anche LISTare e modificare il programma, ammesso che sia scritto in BASIC.

Caricare un Programma Specifico

Per caricare un programma specifico, usate il comando LOAD "nome del programma". Le istruzioni sono le stesse del comando LOAD senza nome, con qualche piccola differenza.

1 Se il programma che volete si chiama "BASE", dovete digitare:

LOAD "BASE"

e premere RETURN

il computer risponde con:

PRESS PLAY ON TAPE

2 Premete PLAY sul Datassette per dire al computer di iniziare la ricerca. Dopo aver cercato sul nastro il programma chiamato BASE dovrebbe apparire il messaggio:

FOUND BASE

LOADING

Lo schermo si spegne mentre il computer "legge" il programma nella sua memoria. Se l'intero nastro finisce prima che appaia il messaggio FOUND, riavvolgete il nastro e provate ancora. Una volta che il computer ha letto tutto il programma il vostro Commodore 16 scriverà:

READY.
█

3 Digitate RUN e Premete RETURN. A questo punto, il Commodore 16 esegue il programma "BASE".

Salvare i Programmi sulle Cassette

Se avete scritto un vostro programma e volete tenerlo per tornarci sopra in futuro potete salvarli (SAVE). Quando salvate un programma, state in effetti registrandolo su un tipo di supporto (cassetta o disco) che vi permetterà di richiamarlo in futuro.
Quando volete salvare un programma su cassetta, fate così:

1 Digitate:

SAVE "nome del programma"

La scelta del nome del programma è liberissima; questo non dovrà però essere più lungo di 16 lettere e/o numeri.

2 Premete il tasto RETURN Il computer visualizza questo messaggio:

PRESS RECORD AND PLAY ON TAPE

3 Premete i tasti RECORD e PLAY sul Datassette. Lo schermo si spegnerà. Quando il programma è stato salvato sullo schermo apparirà la parola "READY".

Esempi:

SAVE "PROVA"
SAVE "GIOCO 1"
"PROVA" e "GIOCO 1" sono i "nomi propri" dei programmi da salvare.

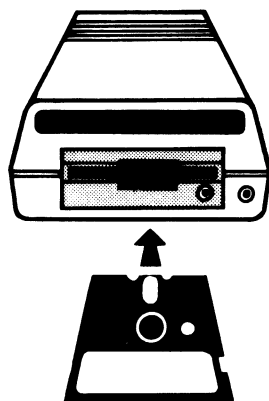
NOTA: Quando salvate un programma su cassetta assicuratevi sempre di come è posizionato il nastro. Fate attenzione in particolare a non salvare un programma proprio all'inizio della cassetta: in molti nastri il primo pezzo non è magnetizzato e la registrazione è impossibile.

DISCHETTI

I dischetti (o floppy disk) sono facili da usare e molto veloci, sempre che li adoperiate con attenzione e con le dovute precauzioni. A differenza di ciò che avviene con le cassette tutto ciò che dovete fare è inserire il disco nel "drive" (l'apposito lettore-scrittore) e digitare sulla tastiera i comandi di lettura o scrittura. I drive non hanno nessun comando diretto (bottoni o altro): solo due spie sul lato frontale: una verde indica se il drive è acceso o spento, mentre quella rossa si accende sia quando il drive sta leggendo o scrivendo qualcosa nel dischetto sia quando qualcosa non sta funzionando. In questo caso la luce lampeggia.

Caricare i Programmi da Disco

1 Assicuratevi che il drive sia acceso

2 Inserite il disco nel drive, (il lato con l'etichetta stampata deve guardare verso l'alto) spingendo fino a quando il margine vicino all'etichetta non è entrato nella fessura. Su di un lato del disco troverete una piccola tacca (coperta a volte con del nastro adesivo); al momento in cui inserite il disco nel drive questa deve trovarsi alla vostra destra.

3 Chiudete lo sportellino di protezione del drive dopo aver inserito il disco.

4 Digitate:

DLOAD "nome del programma" nome specifico del programma da caricare (per risparmiare tempo, potete premere il TASTO DI FUNZIONE 2 e digitare il nome del programma e le seconde virgolette).
5 Premete il tasto RETURN. Il disco inizia a girare e lo schermo mostra:

SEARCHING FOR nome del programma
FOUND nome del programma
LOADING

READY

6 Il vostro software è ora pronto da usare. Digitate RUN e premete il tasto RETURN per far partire il programma.

Se la luce rossa sul disk drive lampeggia dopo che DLOAD ha finito, significa che c'è stato qualche errore. Digitate:

?DS$ (e premete RETURN)

per vedere che errore c'è stato.

Esempi del comando DLOAD:

DLOAD "*"	Carica il primo programma del disco.
DLOAD "PROVA"	Carica dal disco il programma chiamato PROVA.
DLOAD "PRO*"	Carica il primo programma che inizia con le lettere PRO.

Intestare un Dischetto

L'intestazione prepara un disco vergine per l'uso. Ogni disco vergine deve essere formattato prima dell'uso, usando il comando HEADER.

IMPORTANTE: non formattate nessun disco su cui siano registrati programmi o altro, a meno che non vogliate cancellare tutto ciò che vi è inciso. La formattazione cancella tutto ciò che è registrato su un disco.

Il formato per il comando HEADER è:

HEADER"nome del disco"[,Udevice][,Ii.d.],Ddrive

- Il nome che usate è il nome dell'intero disco. Potete dare al disco qualunque nome fino a 16 caratteri.
- Device specifica al computer la periferica in uso (il disk drive invece del Datassette), ed è di solito il numero 8.
- L'i.d (identificatore) è la lettera I seguita da due lettere e/o numeri come I21, IR5, ecc. Date ai dischi il i.d che volete, ma fate attenzione che sia diverso.
- Se avete una unità a dischi doppia dovete specificare il numero di drive con 0 o 1.

ARE YOU SURE? (Siete sicuri?)

Non appena premete RETURN dopo aver dato il comando HEADER, il Commodore 16 chiede ARE YOU SURE? Questo per darvi l'ultimo possibilità di cambiare idea. Se volete formattare il disco, digitate YES o Y e premete RETURN. Se decidete di non formattare il disco, digitate NO o N e premete RETURN.
Ecco alcuni esempi del comando HEADER:

HEADER "LETTERE", I07, D0
HEADER "FINANZA", U8, IS3, D0

Ora che sapete come formattare un disco, siete in grado di usare i dischi per scrivere e salvare i programmi sul vostro Commodore 16 (o su qualunque altro computer Commodore). L'enciclopedia BASIC in fondo a questo manuale dà ulteriori informazioni sul comando HEADER.

Salvare i Programmi sui Dischetti

Quando pensate di poter avere ancora bisogno di un programma che avete scritto, assicuratevi di averlo salvato prima di caricare un altro programma o di spegnere il Commodore 16. Altrimenti lo perderete definitivamente.
Quando modificate un programma salvato, dovete ri-salvarlo se volete tenere la nuova versione. Quando ri-salvate un programma la nuova versione sostituisce quella vecchia. Se volete mantenere sia la vecchia che la nuova dovete salvare la nuova con un nome differente.
Per salvare un programma su disco:

1 Digitate DSAVE "nome del programma"

2 Premete RETURN Il computer visualizza questo messaggio quando il programma è stato salvato:

SAVING 0:"nome del programma"
READY
Esempio:

DSAVE "MIOPROG5" (Il nome del programma può essere lungo fino a 16 caratteri)
Se la luce rossa sul drive lampeggia dopo che DSAVE ha finito, c'è stato qualche errore. Digitate:

?DS$ (e premete RETURN)

Per vedere cosa è successo.

Se cercate di salvare un programma su un disco protetto da scritture (un disco che non accetta i vostri comandi) dovrete poi spegnere e riaccendere l'unità dischetti.

Il Comando Directory

Quando salvate un programma sul disco, il computer tiene un listato di tutti i files salvati su quel disco. Potete visualizzare il listato come un indice per vedere cosa c'è su un dato disco con il comando DIRECTORY

Digitate DIRECTORY e premete RETURN
(o premete il TASTO DI FUNZIONE 3)

Non appena premete RETURN, il Commodore 16 visualizzerà tutto ciò che è presente sul disco.
Potete anche visualizzare una parte dell'indice:

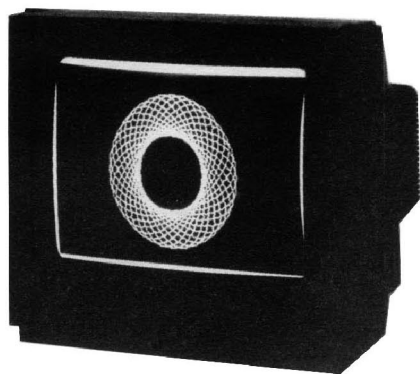
DIRECTORY "GI*"	Lista ogni file che inizia con le lettere GI.
DIRECTORY "*=PRG"	Lista ogni file programma
DIRECTORY "*=SEQ"	Lista ogni file sequenziale

CAPITOLO 4

PRIMI PASSI COL COMMODORE 16

- Lo schermo video
- Stampa in negativo e cambio colori
- Il primo passo
- Correggere gli errori
- Finestre video

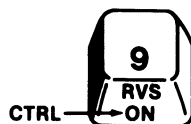
INTRODUZIONE

Lo scopo di questo capitolo è quello di iniziare a farvi conoscere le caratteristiche e le capacità del vostro Commodore 16, e di farvi fare i primi passi verso la programmazione del vostro computer.

Lo Schermo Video

Tutto ciò che avviene nel vostro Commodore 16 viene mostrato sullo schermo. Qualunque cosa scriviate, qualunque cosa il computer debba dirvi a proposito di ciò che avete scritto, qualunque programma carichiate... sarà tutto mostrato sullo schermo video, sia esso un TV color o in B/N, o un monitor a colori Commodore. Le schermate sono comunque sempre lunghe 40 caratteri, (il che significa che potete battere 40 caratteri dal bordo sinistro dello schermo al destro), e da 25 righe. Uno schermo interamente riempito contiene 1000 (25 x 40) caratteri. Lo schermo del Commodore 16 è formato da tre elementi: i caratteri (le lettere, i numeri e i simboli grafici), lo sfondo, e il bordo.
Quando accendete il computer per la prima volta, i caratteri sono neri, lo sfondo è bianco e il bordo è viola. Avete già visto come cambiare il colore dei caratteri usando il tasto SHIFT col CTRL o C= nel capitolo 2. Potete fare i caratteri di 16 colori diversi in questo modo. Ci sono alcune altre cose che potete fare per rendere più interessante la stampa dei caratteri.

Stampa in Negativo e Cambio del Colore

Non solo potete cambiare il colore al cursore e ai caratteri che digitate (e che il computer visualizza) sullo schermo video, potete anche mettere i caratteri in negativo. Ciò significa che se avete caratteri neri su sfondo bianco e azionate la stampa in negativo (digitando CTRL e 9), tutto ciò che appare sullo schermo è messo in negativo: i caratteri sono bianchi, mentre lo sfondo dietro questi caratteri è nero. Potete provare a cambiare il colore dei caratteri e ad usare la stampa in negativo, seguendo queste istruzioni:

1 Tenete premuto il tasto CTRL e premete il tasto RVS ON

2 Rilasciate i tasti e tenete premuta la barra spaziatrice.

3 Tenete premuta la barra spaziatrice: sullo schermo vedrete allungarsi una linea dello stesso colore delle lettere. Se la linea raggiunge la fine della riga continua alla riga seguente.

4 Rilasciate la barra spaziatrice (ma non premete il tasto RETURN).

5 Tenete premuto il tasto CTRL e premete uno dei tasti del colore (non un colore che c'è già sullo schermo). Non appena fate ciò, il cursore diventa del colore del tasto che avete premuto.

6 Premete ancora la barra spaziatrice: il vostro computer traccerà una linea nel colore desiderato.
Per cambiare ancora colore schiacciate CTRL e C= e i tasti per il colore e premete la barra spaziatrice.

7 Spegnete la stampa in negativo tenendo premuto CTRL e premendo il tasto RVS OFF. Anche il premere il tasto RETURN distabilita la stampa in negativo.

Breve parentesi: Potete digitare lettere, numeri e punteggiatura, tutto in negativo. Le lettere in negativo permettono di fare eccellenti intestazioni. Potete anche utilizzarle per mettere in evidenza particolari parole e numeri; lo stesso vale per i simboli grafici. Per dividere in bande colorate il vostro schermo selezionate il colore che preferite e digitate il simbolo — (SHIFT e E) per tutta la linea. Quindi abilitate la stampa in negativo; riempite la prossima riga con lo stesso simbolo, solo in negativo. Sulla linea successiva premete il tasto SHIFT e R (ancora in negativo). Poi disabilitate la stampa in negativo e riempite l'ultima riga con SHIFT R (non più in negativo). Ora il vostro schermo è diviso in bande del colore che avete scelto; forse è un modesto risultato, ma dà al vostro schermo un certo non so che...

IL PRIMO PASSO

Cosa è un programma? Un esperto programmatore potrebbe dirvi che un programma è una serie ordinata di istruzioni organizzata in modo da far fare al computer una serie di operazioni al fine di ottenere un determinato risultato. Ciò che avrete compreso da tutto questo è di non chiedere mai ad un programmatore esperto cosa è un programma.
Ciò che segue, sebbene sia lungo due righe, dà l'idea di cosa sia in effetti un programma. Digitatelo esattamente come appare. Non dimenticate i numeri all'inizio di ogni linea, dato che sono i numeri di linea; essi dicono al computer in che ordine leggere ed eseguire le istruzioni del programma. Ricordatevi di premere il tasto RETURN alla fine di ogni linea.

10 PRINT "PRIMA PROVA"	Questa linea dice al computer di stampare le parole PRIMA PROVA sullo chermo.
20 GOTO 10	Questa linea dice al computer di tornare alla riga 10 a stampare PRIMA PROVA ancora.
RUN	Quando digitate questo comando e premete RETURN il computer esegue il programma.

Nota: attenzione quando si programma un computer alla lettera O ed al numero 0 (zero, che come noterete è sbarrato, proprio per non confonderlo con la lettera dell'alfabeto). I due segni sono completamente diversi.

Premete il tasto RUN/STOP per fermare il programma.
Perché il computer ha stampato PRIMA PROVA così tante volte? PRINT è un comando che dice al computer di stampare sullo schermo ciò che appare tra virgolette. Quando il computer legge quella linea esegue quel comando. GOTO dice al computer di tornare alla linea 10 ad eseguire il comando (stampa PRIMA PROVA). Ogni volta che il computer arriva alla linea 20, esegue il comando GOTO, e torna quindi sempre alla linea 10. Questa ripetizione è chiamata "LOOP". Il tasto RUN/STOP interrompe qualunque programma il computer stia eseguendo.

Nota: Volete rallentare questo programma (o qualunque altro) senza fermarlo? Basta che teniate premuto il tasto C= .

Ora digitate ciò che segue, premendo RETURN dopo ogni linea:

NEW — Questa linea dice al computer di cancellare l'ultimo programma e di prepararsi per uno nuovo.

Il computer risponde:

READY — Il computer vi avverte che è pronto (READY) per nuovi comando o programmi.

10 PRINT "(premere CTRL e RVS ON) COMMODORE (premere CTRL e RVS OFF) 16"

Notate cosa succede quando premete RVS ON e RVS OFF dentro le virgolette. Appaiono sullo schermo come caratteri in negativo. Ora digitate RUN e premete RETURN per vedere il programma. Il Commodore 16 legge questi caratteri in negativo come gli è stato detto di fare. Lo stesso principio si rivela vero (un carattere in negativo che appare sullo schermo come rappresentazione di ciò che avete digitato) per fare anche altre cose all'interno di un PRINT, come spostare il cursore e cambiare i colori, quando lanciate (RUN) un programma.

Ora riprovate la stessa linea, ma sostituite RVS ON e RVS OFF con FLASH ON e FLASH OFF (tasti < e >):

10 PRINT "(premere CTRL e FLASH ON) COMMODORE (premere CTRL e FLASH OFF) 16"

Una volta lanciato questo programma stampa COMMODORE 16 come il precedente. Ma la parola Commodore lampeggia continuamente, anche se il programma è già stato eseguito.

Inserire i Comandi

Come mai a volte i comando possono essere eseguiti direttamente (ad esempio RUN, LIST, DSAVE), e altre volte deveno essere inseriti in strutture come "10 PRINT..."; RETURN; RUN; RETURN? Tutti questi comandi sono basati su un linguaggio fatto di termini che il computer può capire. Il più semplice (quello installato sul Commodore 16) si chiama BASIC. Il Commodore 16 capisce una particolare versione chiamata Commodore 3.5 BASIC che ha due "modi". I termini (o parole chiave) del BASIC sono il cuore di entrambi i modi. Il primo tipo, MODO IMMEDIATO, dice al computer di eseguire il comando BASIC immediatamente. Basta digitare il comando e alla pressione di RETURN questo sarà eseguito. Questo è anche conosciuto come MODO DIRETTO. L'alto od è conosciuto come MODO PROGRAMMA o MODO INDIRETTO. Il MODO PROGRAMMA comporta la presenza dei numeri di linea, ed ogni linea contiene comandi BASIC. L'intero programma è eseguito quando digitate il comando RUN, che fa eseguire i comandi del programma secondo il numero di linea (a partire dalla linea col numero più basso ed in ordine crescente). Ovviamente, il MODO PROGRAMMA è quello che si usa per scrivere i programmi per il computer.

CORREGGERE GLI ERRORI

Gli errori sono comuni con i computers, ma devono essere trovati ed eliminati se si vuole che il programma giri correttamente. Il tasto HELP serve proprio a questo. Uno degli errori più comuni – l'errore di battitura – può pregiudicare il funzionamento di un programma. Non solo la programmazione migliorerà le vostre doti di dattilografo, ma vi farà anche fare grandi progressi nell'individuazione e nella correzione degli errori di battitura.
Vi sono diversi modi per trovare gli errori. Ricordate che ogni volta che fate qualche correzione dovete sempre confermarla al computer premendo RETURN

1. POTETE MODIFICARE UNA LINEA scrivendoci sopra.

Usate i tasti di cursore per muovervi al punto della linea che volete cambiare. Ora digitate semplicemente sopra a quello che volete cambiare.

ESEMPIO:

10 PRINT "SONO LE DUE"

Se volete cambiare l'ora a TRE, spostate il cursore sopra la D di DUE.

10 PRINT "SONO LE DUE"

Basta ora digitare TRE sopra a DUE e premere RETURN

10 PRINT "SONO LE TRE".

Nota: Quando lavorate con le linee numerate di un programma, non è necessario che vi portiate alla fine della linea per premere RETURN. Il Commodore 16 ricorda l'intera linea anche se premete RETURN nel mezzo della linea.

2. POTETE INSERIRE DEGLI SPAZI IN UNA PAROLA O IN UNA LINEA con il tasto INST (premere SHIFT assieme a INST/DEL).

Tenete premuti i tasti fino a quando avete inserito gli spazi necessari. (Notate che il cursore rimane fermo mentre gli spazi sono inseriti alla sua destra). Ora potete digitare ciò che dovevate inserire.

10 PRINT "CORE"

Per cambiare la scritta in COMMODORE, spostate il cursore sulla "O" e premete i tasti SHIFT e INST fino a quando avete inserito abbastanza spazi.

10 PRINT "CO ORE"

Ora inserite le altre lettere:

10 PRINT "COMMODORE"

3. POTETE CANCELLARE CARATTERI ED ELIMINARE SPAZI col tasto DEL (si ottiene la funzione DEL premendo tasto INST/DEL da solo). Questo tasto cancella i caratteri o gli spazi che si trovano immediatamente alla sinistra del cursore.

10 PRINT "COSTO GIORNALIERO"

Potete cambiare la frase in COSTO ORARIO spostando il cursore sulla A di GIORNALIERO, premendo cinque volte il tasto DEL e scrivendo ORARIO.

10 PRINT "COSTO GIORNALIERO"

e premete INST/DEL cinque volte

10 PRINT "COSTO ALIERO"

Digitate ORARIO al posto di ALIERO e premete RETURN

4. **POTETE RI-DIGITARE UNA LINEA** in qualunque momento, anche dopo aver lanciato il programma. Il Commodore 16 sostituisce automaticamente la nuova linea alla vecchia quando premete RETURN. La vecchia linea rimane ancora sullo schermo, ma il computer la ignora. Quando avete due linee con lo stesso numero, il Commodore 16 usa solo quella immessa per ultima. Per esempio, in un breve programma che usa il comando COLOR per cambiare il colore dello sfondo, vi potrebbe essere un errore:

10 COKOR 0,3 errore

20 PRINT "COMMODORE 16"

Premete il tasto RETURN fino a portarvi su una linea bianca, e ri-digitate la linea 10 correttamente:

10 COLOR 0,3

Ora la prima linea 10 è stata sostituita dalla seconda. Potete averne la conferma digitando LIST, che visualizza linea per linea il LISTato del vostro programma così come è memorizzato nella memoria del computer. Quando listate un programma tutte le linee appaiono in ordine numerico, mentre le linee sostituite non appaiono:

LIST RETURN

Sullo schermo appare:

10 COLOR 0,3

20 PRINT "COMMODORE 16"

Potete vedere cosa fa il comando COLOR in questo esempio. Per una spiegazione più completa, vedete il Capitolo 6, o cercate COLOR nella Enciclopedia Basic.
Sostituire le linee in un programma è anche un buon modo per fare esperimenti col vostro computer. Quando sostituite una linea, non è necessario che quella nuova sia simile a quella vecchia. Per esempio, anziché correggere l'ortografia di COLOR, potete digitare questa linea:

10 PRINT "COMMODORE 4 PER COMMODORE 4 ="

Ora lanciate (RUN) il programma e guardate cosa succede.

4. **POTETE CANCELLARE UNA LINEA INUTILE** digitando semplicemente il suo numero di linea e premendo RETURN. Il computer ignora la linea anche se continua ad apparire sullo schermo. Digitate LIST per avere il listato del programma per assicurarvi che la linea è sparita dal programma.

```
10 PRINT "COMMODORE 4 PER COMMODORE 4 ="
20 PRINT "COMMODORE 16"
```

Ora digitate

10 e premete RETURN
LIST

Il risultato dovrebbe essere

```
20 PRINT "COMMODORE 16"
```

CANCELLARE LO SCHERMO

Può capitare che lo schermo sia troppo affollato, o che vogliate riorganizzare ciò che avete digitato cancellando lo schermo. Vi sono diversi modi per fare ciò, lasciando il programma in memoria o cancellando sia memoria che schermo. Un modo semplice per cancellare lo schermo consiste nel tenere premuta la barra spaziatrice fino a quando vengono cancellati tutti i 1000 caratteri presenti sullo schermo (ci vorranno un paio di minuti). Se avete letto la Guida per l'Utente finora, è probabile che abbiate anche la pazienza di cancellare lo schermo in questo modo; ma anche l'intelligenza di capire che ci deve essere un modo migliore. Ecco alcuni modi migliori:

1. Premere i tasti SHIFT e CLR/HOME assieme. In questo modo si cancella lo schermo e si riposiziona il cursore nell'angolo in alto a sinistra (la posizione "home").

2. Premere il PULSANTE DI RESET. Questo resetta la macchina, cioè la fa ritornare nelle stesse condizioni in cui era all'accensione, cancellando la memoria e facendo apparire il messaggio di presentazione.

3. Digitate:

SCNCLR e premete RETURN

Questo è un comando del BASIC che il computer interpreta come istruzione per cancellare lo schermo.

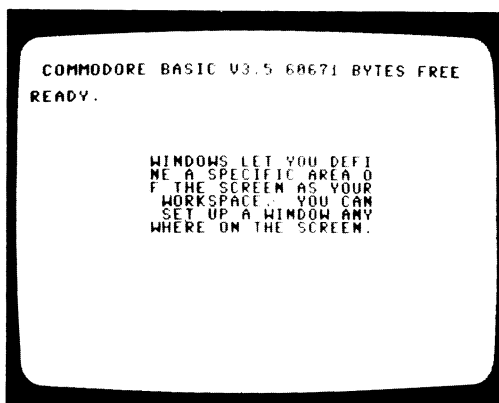
4. Digitate:

NEW e premete RETURN

Questo è un comando del BASIC che dice al computer di cancellare la sua memoria. Il "16" del nome "Commodore 16" si riferisce alla quantità di spazio che il computer possiede per immagazzinare i programmi, conosciuta come memoria. Quando cancellate lo schermo con i metodi 1 e 3, cancellate i caratteri presenti sullo schermo, ma ciò che è nella memoria del computer non viene toccato. Questo significa che avete meno spazio per i programmi. I metodi 2 e 4 cancellano ciò che è contenuto nella memoria del computer, in modo da avere l'intero spazio in memoria disponibile per nuovi programmi.

FINESTRE SULLO SCHERMO

```
COMMODORE BASIC V3.5 60671 BYTES FREE
READY.

                WINDOWS LET YOU DEFI
                NE A SPECIFIC AREA O
                F THE SCREEN AS YOUR
                 WORKSPACE.  YOU CAN
                 SET UP A WINDOW ANY
                WHERE ON THE SCREEN.
```

Le finestre (windows) vi permettono di definire una specifica area di schermo come area di lavoro. Tutto ciò che digitate (le linee che digitate, LISTati di programmi, ecc.) dopo aver costituito una finestra appare all'interno dei confini della finestra, senza disturbare lo schermo fuori dall'area. Si possono creare finestre in qualunque punto dello schermo.

Per costituire una finestra:

1 Spostate il cursore nella posizione in cui volete l'angolo superiore sinistro della finestra.

2 Premete il tasto ESC e la lettera T.

3 Spostate il cursore nella posizione in cui volete l'angolo inferiore destro della finestra.

4 Premete ESC seguito da B, ed ecco la finestra.

Tutto ciò che è stampato sullo schermo è confinato all'area che avete definito. Per cancellare la finestra premete HOME due volte. La finestra viene cancellata, e il cursore viene posizionato in alto a sinistra.
Potete manipolare la finestra ed il testo al suo interno usando il tasto ESC. Funzioni di editing, come inserimento e cancellazione testo, scrolling, e cambio delle dimensioni di una finestra, possono essere ottenute premendo ESC seguito da un altro tasto. Per usare un comando specifico, premete il tasto appropriato dopo aver premuto ESC.

TASTO	FUNZIONE
A	Inserimento automatico
B	Posiziona l'angolo inferiore destro di una finestra (alla attuale posizione del cursore)
C	Cancella l'inserimento automatico
D	Cancella la riga ove si trova il cursore
I	Inserisce una riga
J	Il cursore si sposta all'inizio della linea dove si trova
K	Il cursore si sposta alla fine della linea dove si trova
L	Attiva lo scrolling
M	Disattiva lo scrolling
N	Ritorna alle normali dimensioni dello schermo
O	Cancella i modi inserimento, virgolette, negativo e lampeggiante
P	Cancella tutto fino all'inizio della linea
Q	Cancella tutto fino alla fine della linea
R	Riduce le dimensioni dello schermo
T	Fissa l'angolo superiore sinistro della finestra
V	Scrolla lo schermo verso l'alto
W	Scrolla lo schermo vero il basso
X	Cancella la precedente funzione di ESCape

CAPITOLO 5

NUMERI E CALCOLI

INTRODUZIONE

Non dovete essere grandi matematici per capire e utilizzare le capacità matematiche del vostro Commodore 16. Oltre alle semplici operazioni come somma, sottrazione, moltiplicazione e divisione potete calcolare funzioni avanzate come radici quadrate e seni. Utilizzerete numeri fino a 38 cifre, o maggiori con la notazione esponenziale. Potrete eseguire calcoli direttamente o come parte di un programma.

NUMERI E OPERAZIONI

Potete usare il computer come una semplice calcolatrice. Oltre agli usuali simboli per le operazioni + e -, il Commodore 16 usa il simbolo * per la moltiplicazione e il simbolo / per la divisione e le frazioni. (I computers usano il simbolo * invece del x per la moltiplicazione poiché il computer non può riconoscere la differenza tra la lettera x e il simbolo matematico x). Potete usare questi operatori sia in modo immediato (senza numero di linea) che in linee di programma. Digitate i numeri e gli operatori in un comando PRINT SENZA LE VIRGOLETTE se volete che il computer calcoli il risultato. Se i numeri e gli operatori sono tra virgolette (come in PRINT "2 + 2"), il computer stamperà esattamente quello sullo schermo, senza eseguire la somma. Potete far sommare due più due digitando PRINT 2+2.

Operatori matematici Basic

Addizione	+
Sottrazione	−
Divisione e frazioni	/
Moltiplicazione	*
Elevazione	↑

Operatori relazionali Basic

Maggiore di	>
Minore di	<
Uguale	=
Maggiore o uguale	>=
Minore o uguale	<=

Nota: Il computer non accetta il puntino (come separatore per le migliaia. Per esempio, dovete digitare 30359 invece di 30.359.

Frazioni e Decimali

Potete scrivere una frazione in questo modo: .5 ripeto: .5
o in questo altro: 1/2. Così il computer esegue la divisione.
Se mettete una frazione in una istruzione PRINT, il risultato è sempre espresso da un numero intero o decimale. Per esempio:

PRINT 139/493 + 5 RETURN
5.28194726

Ecco un esempio che usa pi greco (3.141592651 che rappresenta il valore di una circonferenza rispetto al suo diametro. Usate questo valore semplicemente premendo il tasto π, (C= e =)

PRINT π/374 RETURN
8.39998036E-03

Notazione Scientifica

Cosa voleva dire il computer con la parte E-03 del risultato precedente? Il computer visualizza i numeri decimali da -999999999 a +9999999999 come numeri normali. I numeri al difuori di questo intervallo (con più di nove cifre) sono automaticamente mostrati con la notazione scientifica. Anche voi potete inserire i numeri in questa maniera e il computer non avrà alcun problema a leggerli correttamente. La notazione scientifica è spesso utile, dato che permette al computer di visualizzare grandi numeri con poche cifre.
Ecco come il numero 198, 505,472 (le virgole separano le migliaia) sarebbe scritto con la notazione scientifica:
solo UNA cifra è mostrata alla sinistra del punto decimale
1.98505478E+8 questo numero (+8) è il numero che rappresenta di quante cifre deve essere spostato il punto decimale
Per un numero minore di uno con numerose cifre dopo la virgola, il secondo numero sarebbe un - anziché un + che indica che il punto decimale è spostato a destra.

Per esempio:

.0003359= 3.359E-4

Altri esempi:

20 = 2E+1
10500 = 1.05E+5
.0666 = 6.66E-2

ESEGUIRE I CALCOLI

Per far eseguire un calcolo al vostro computer scrivete "PRINT" e poi i numeri, come segue:

```
10 PRINT 1+2, 2-1
20 PRINT 2*2, 4/2
RUN

3       1
4       2
```

Per la prima volta PRINT non ha stampato esattamente ciò che avevate digitato. Al contrario, il computer ha risolto il calcolo e ha stampato solo il risultato. Tutto ciò che dovete fare per usare PRINT per calcolare è di omettere le virgolette.
Ora provate questo:

```
NEW
10 PRINT "2001/2010"
20 PRINT 2001-17
RUN
2001/2010
 1984
```
qui viene lasciato uno spazio per il segno del risultato

Dato che la linea 10 è tra virgolette, il computer ha semplicemente stampato ciò che appare tra le virgolette.
Notate che non viene lasciato alcuno spazio per il segno del numero dalla linea 10, mentre lo è dalla linea 20. Ora spostate il cursore sulla linea 10 e cambiatela così:

```
10 PRINT "2001/2010="; 2001/2010
RUN
2001/2010= .995522388
 1984
```
non dimenticate il punto e virgola

questo spazio è lasciato per il segno del risultato

il risultato per la linea 20 rimane lo stesso

Se volete stampare sia il problema che il risultato dovete digitarlo due volte: una volta tra virgolette e una volta al di fuori.

Ordine di Calcolo

Potete fare più di un calcolo sulla stessa linea. Provate a digitare questo:

PRINT 200+50/5

È il risultato che vi aspettavate? Provate questo:

PRINT (200+50)/5

Il Commodore 16 risolve ~~sempre per~~ prima quella parte di problema racchiusa tra parentesi. Potete anche mettere delle parentesi all'interno di altre: 36*(12+(A/3). Il contenuto della parentesi più interna è calcolato per primo. Qualche volta è una buona idea mettere i numeri negativi tra parentesi per maggior chiarezza. Per esempio, se volete moltiplicare 45 per -5, digitatelo così:

45*(-5)

Il computer comunque lo capisce con o senza parentesi.

PRECISAZIONI SULLA STAMPA SULLO SCHERMO

Avete probabilmente notato che in certi programmi avete dovuto digitare delle virgole in certi posti, mentre in altri dei punti e virgola. Il computer interpreta virgole e punto e virgola come istruzioni per la spaziatura durante la stampa. L'effetto della punteggiatura sulla spaziatura nelle istruzioni PRINT funziona allo stesso modo sia con il testo tra virgolette (chiamato "stringa di testo") che con i numeri calcolati. Provate a digitare questo breve programma:

```
NEW
10 ?PRINT"O","K"
20 PRINT"O";"K"
```

Notate che la punteggiatura appare FUORI dalle virgolette.
Quando lanciate (RUN) il programma, lo schermo appare così:

O K	risultato della linea 10
OK	risultato della linea 20

Se la linea 10 e la linea 20 sono quasi identiche, perché c'è così tanta differenza in ciò che stampano sullo schermo? La sola differenza è dovuta alla punteggiatura tra le lettere.
Quando usate una virgola per separare più messaggi in un comando PRINT, i messaggi sono stampati intervallati da alcuni spazi. Quando usate il punto e virgola, i messaggi sono stampati uno accanto all'altro.
Come ricordate, lo schermo del vostro computer ha 40 colonne. Queste sono divise in quattro aree da 10 caratteri, chiamate ZONE DI STAMPA. Quando usate una virgola per separare i messaggi stampati, il Commodore 16 stampa il primo messaggio nella prima zona di stampa, il secondo nella seconda zona ecc. Questo comando funziona come il tabulatore in una macchina da scrivere.

```
ZONA DI STAMPA 1    ZONA DI STAMPA 2    ZONA DI STAMPA 3    ZONA DI STAMPA 4

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 >COLONNE
O                K
```

Se provate a stampare più di quattro messaggi separati da virgole, il vostro computer va automaticamente alla linea seguente per stampare. Per esempio:

```
PRINT"A","B","C","D","F"
```

Le lettere sullo schermo vengono così spaziate:

		1	11	21	31	40	COLONNA
RIGA	1	A	B	C	D		
	2	E	F				

Quando usate il punto e virgola per separare più messaggi in un comando PRINT, il Commodore 16 ignora le zone di stampa e stampa tutti i messaggi uno dopo l'altro. Per esempio il comando:

PRINT"A";"B";"C";"D";"E";"F"

stampa

ABCDEF

Se il primo messaggio è lungo 12 lettere ed è separato dal secondo da una virgola, ecco cosa succede:

PRINT"ABCDEFGHIJKL","M"

Stampa questo:

ZONA 1   ZONA 2   ZONA 3
ABCDEFGJHIKL   M

NOTA: Qualche volta digiterete delle linee particolarmente lunghe, come questa:

10 PRINT "MI PIACE COME TOCCHI LA MIA TASTIERA. CAPITI QUI SPESSO?"

Anche se andate oltre la lunghezza di una linea, continuate a digitare: il Commodore 16 automaticamente si sposterà alla riga seguente continuando a stampare fino a quando avrete finito. Potete digitare fino a 80 caratteri in una linea di programma (fino a due linee complete da 40 caratteri).
Ora provate a lanciare il programma di prima. Il messaggio è stampato su due righe. Quando la linea è più lunga di una linea di schermo, il Commodore 16 automaticamente la fa continuare alla linea seguente. Il vostro computer considera finita una linea quando voi premete RETURN, non quando arrivate al limite destro dello schermo.
Vi abituerete a questo con l'uso del vostro Commodore 16.

USARE LE VARIABILI

L'esempio 36*(12+(A/3)) vi mostra una delle più potenti caratteristiche di un computer. Quando usate una lettera al posto di un numero in un problema matematico, state usando una VARIABILE. Una variabile rappresenta un valore:

```
10 A=3
20 PRINT "TOTALE:";A*4
```

Se lanciate questo programma, sullo schermo apparirà:

TOTALE: 12

Ci sono tre tipi di variabili che potete usare:

TIPO	SIMBOLO	DESCRIZIONE	ESEMPI	ESEMPIO DI VALORI
Reale		numeri reali o interi	X,AB,T4	23.5,12,1.3E+2
interi	%	numeri interi	S%,A1%	15,120,1478
stringhe alfanumeriche	$	lettere, numeri tutti i caratteri tra virgolette	S$,MS$	"TOTALE:" "GIORNO 1"

Ogni volta che volete che una variabile rappresenti un numero intero, questa deve terminare col simbolo %. Una variabile alfanumerica deve terminare col simbolo $. Una variabile senza il simbolo $ o % viene interpretata come variabile reale mobile. Le variabili intere sono un sottoinsieme di quelle in virgola mobile: sono numeri senza punto decimale.
Assicuratevi di usare sempre il giusto tipo di variabile. Se provate ad assegnare una parola ad una variabile intera, il vostro programma non funzionerà. Questo programma vi mostra quali variabili possono o non possono essere utilizzate in determinate situazioni.

```
10 PRINT "INSERITE UN NUMERO"

20 INPUT X%

30 PRINT "IL NUMERO È";X%

40 PRINT "TUTTO È ANDATO BENE"

50 END
```

vi chiede di inserire un numero per la variabile

Quando date il RUN provate ad introdurre questi valori (uno ogni volta che date il RUN) quando ve lo viene richiesto e guardate cosa succede:

UN QUINTO
.043
10

FUNZIONI NUMERICHE

Nel linguaggio BASIC del vostro computer sono incluse anche le funzioni matematiche, come quelle che si trovano nella maggior parte delle calcolatrici scientifiche (seno, coseno, tangente ecc.).
La maggior parte delle funzioni matematiche possono essere usate digitando il nome della funzione seguito dall'argomento della funzione posto tra parentesi, in questo modo:

```
FUNZIONE(X)
```

Per esempio se volete calcolare il seno di una variabile dovrete scrivere:

```
PRINT SIN(X)
```

con la X che rappresenta il numero che gli volete dare.
Potete anche includere una di queste funzioni in una linea di programma, come mostra il seguente esempio:

```
10 FOR X=1 TO 5
20 PRINT "LA RADICE QUADRATA DI";X;"È";SQR(X)
30 NEXT X
```

C'è un elenco completo delle funzioni numeriche nella "Enciclopedia Basic 3.5".

Funzioni Definite dall'Utente

Una maniera di utilizzare le capacità matematiche del vostro computer è quella di creare funzioni definite dall'utente. Le funzioni definite dall'utente sono estremamente utili nei calcoli e facili da implementare usando il Commodore 16; vi danno la possibilità di programmare una formula lasciando al computer il compito di inserire i valori da calcolare. Questa può essere usata per molti usi.
Ecco una linea di programma che utilizza una funzione definita dall'utente per calcolare il valore di una secante:

```
10 DEF FNS(X)=1/COS(X)
```

Questa linea di programma calcola il valore della secante di un numero qualsiasi rappresentato dalla X. FNS è il nome della funzione definita da questa linea.
L'appendice G contiene una tabella delle funzioni matematiche non incluse nel linguaggio BASIC del computer.
Le funzioni definite dall'utente vi fanno risparmiare spazio in memoria quando dovete usare una funzione più di una volta, e rendono i vostri programmi più facili da leggere e capire.

CAPITOLO 6

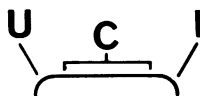
COLORI
E
GRAFICA

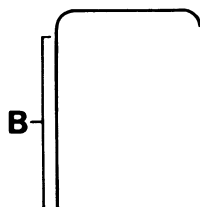
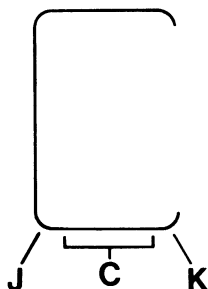
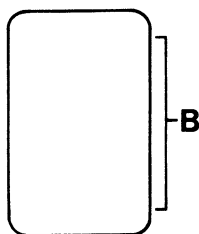
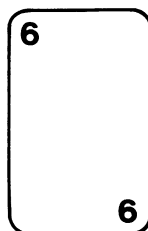
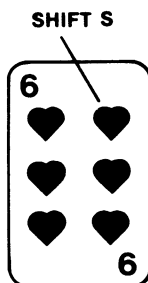
- Caratteri grafici
- Animazione di caratteri
- Controllo del colore
- Grafica ad alta risoluzione
- Punti, linee, ed etichette
- Quadrati, cerchi, poligoni e riempimento di aree
- Grafica multicolore

CARATTERI GRAFICI

Forse vi ricordate ancora (era nel capitolo 2) che ogni tasto alfabetico contiene due diversi caratteri grafici, così come i simboli ? , -, *, e il tasto della sterlina inglese per un totale di 62 caratteri grafici diversi. Per stampare questi caratteri grafici bisogna tenere premuto il tasto SHIFT o il tasto C= insieme al tasto del simbolo grafico voluto.
Quando il Commodore 16 non è in modo macchina da scrivere, tenete premuto il tasto SHIFT e premete un tasto alfabetico per ottenere il carattere grafico che si trova sulla destra del tasto. Questi caratteri comprendono i semi delle carte da gioco, una pallina piena e una vuota e un set di linee e di caratteri di giunzione che potete usare per disegnare varie figure sullo schermo.
Premendo il tasto C= insieme ad un tasto alfabetico si ottiene il carattere grafico disegnato sulla parte sinistra del tasto. I caratteri grafici sulla parte sinistra sono un assortimento di barre, quadratini, linee, molto utili per disegnare grafici e diagrammi.

Disegnamo una Carta da Gioco

Ecco un esempio per aiutarvi ad avere un'idea più chiara di come usare i caratteri grafici per creare dei disegni.
Seguendo queste istruzioni potrete creare una carta da gioco, in questo caso un sei di cuori.
Per prima cosa cambiate il colore del cursore facendolo diventare rosso. Tenete premuto il tasto CTRL assieme al tasto "3" per cambiare il colore del cursore.

Ora è il momento di disegnare il bordo superiore della carta: premete il tasto SHIFT LOCK

in modo che resti premuto. Premete il tasto "U" seguito dal tasto "C" per cinque volte e quindi il tasto "I".

Per disegnare la parte sinistra: lasciando abbassato il tasto SHIFT LOCK andate allo spazio giusto al disotto del bordo curvo (dove avete premuto il tasto "U" nel passo precedente). Premete il tasto "B", spostate poi il cursore nello spazio giusto al disotto del segmento che avete appena tracciato e premete ancora "B". Fate questo fino a quando avete premuto il tasto "B" per sette volte. (L'uso dei tasti di cursore sarà dapprima difficoltoso; ma migliorerà con la pratica).

Per disegnare il lato inferiore: come per il bordo superiore premete il tasto "J" una volta e poi premete il tasto "C" cinque volte, concludendo con il tasto "K".

Per disegnare il lato destro: disegnate quest'ultimo come avete fatto per il lato sinistro premendo il tasto 'B' per un totale di sette volte (portando il cursore nei posti giusti).

Rilasciate il tasto SHIFT LOCK premendolo un'altra volta. Premete il tasto "6" nell'angolo in alto a sinistra (alla distanza di uno spazio sia in basso che a destra dello spigolo della carta). Portate il cursore nell'angolo in basso a destra della carta uno spazio sopra e a sinistra dell'angolo della carta e premete un'altra volta il tasto "6".

Ora per disegnare i cuori, sei per la precisione, premete di nuovo il tasto SHIFT LOCK. Scendete e spostatevi a destra di uno spazio rispetto a dove avete scritto il 6. Premete il tasto "S", premete lo spazio e ancora il tasto "S". Scendete col cursore di due linee e ripetete la sequenza "S-SPAZIO-S". Aggiungete gli ultimi due cuori due righe al disotto. Ora avete un sei di cuori uguale a quello delle carte da gioco. Se volete allenarvi, provate a creare un paio di carte da soli.

Nota: Quando lo SHIFT LOCK è premuto, potete premere RETURN senza ottenere un SYNTAX ERROR come risposta del computer. Anche se su quella linea erano presenti dei caratteri non riconosciuti dal computer come comandi, il computer interpreta il RETURN come istruzione per tornare all'inizio della linea quando il tasto SHIFT è premuto. Il computer non cerca di leggere ed interpretare ciò che è stato digitato come comando BASIC quando lo SHIFT o lo SHIFT LOCK è premuto.
Potete usare i simboli grafici per migliorare la stampa. Per esempio, ecco come sottolineare una parola o una linea. Spostate il cursore sotto alla linea che volete sottolineare; poi tenete premuto il tasto C= e il tasto "T" per stampare fino a quando avete completato la sottolineatura.
Lo scopo di questa sezione è di mostrarvi come si possono manipolare i simboli grafici del Commodore 16 per creare diverse figure in un modo più pratico. Oltre ai 62 caratteri grafici disponibili, potete anche usare i negativi di questi caratteri. Ora che avete una buona idea di come si usano i diversi simboli grafici per creare delle figure, dovreste provare a fare esperimenti da soli, per vedere come ve la cavate.

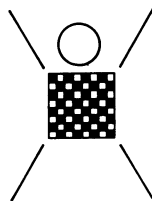
ANIMAZIONE DI CARATTERI

I film sono sequenze di immagini ferme. Ogni immagine è leggermente diversa da quella apparsa prima. Il proiettore mostra ogni immagine per un tempo molto breve, poi passa alla successiva.
L'animazione col computer funziona allo stesso modo. Prima il computer disegna una immagine: poi la cambia lentamente. Il Commodore 16 è abbastanza veloce per spostare gli oggetti dolcemente per lo schermo nei vostri giochi e programmi. Un film è animato al ritmo di 30 immagini al secondo. I cambiamenti sono abbastanza rapidi da ingannare l'occhio e creare l'illusione di movimento. Potete ottenere questa illusione sul computer con un programma che disegna un'immagine, aspetta una frazione di secondo e poi mostra una nuova figura. Per creare immagini col computer dovete usare il comando PRINT con i caratteri grafici. Il più semplice tipo di animazione comporta l'alternarsi di due caratteri per ottenere il movimento.
Questo programma simula l'animazione alternando il cerchio (SHIFT & Q) e il cuore (SHIFT & S).

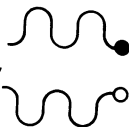
Nota bene: Ogni volta che ci riferiamo ai tasti SHIFT e C= , questi devono essere premuti ASSIEME al carattere che segue, dato che non accade nulla premendoli da soli.
Ricordate di digitare NEW e di premere RETURN prima di inserire ogni programma e premete RETURN al termine di ogni linea di programma.

```
10 PRINT "HOME SHIFT Q"
20 FOR L=1 TO 100
30 NEXT L
40 PRINT "HOME SHIFT S"
50 FOR M=1 TO 200
60 NEXT M
70 GOTO 10
```

Per fermarlo, premete il tasto RUN/STOP.
Per ottenere effetti più interessanti potete costruire un piccolo disegno con alcuni caratteri grafici, poi cambiare pochi caratteri lasciando gli altri invariati allo stesso posto. Così otterrete l'effetto di una parte di un oggetto grande in movimento, come nel seguente programma:

```
10 PRINT " HOME SHIFT M SHIFT W SHIFT N "
20 PRINT " SPACE C= + SPACE "
30 PRINT " SHIFT N SPACE SHIFT M "
40 FOR L=1 TO 300: NEXT L
50 PRINT " HOME SPACE SHIFT W SPACE "
60 PRINT " C= T C= + C= T "
70 PRINT " SPACE C= G C= G "
80 FOR L=1 TO 300: NEXT L
90 GOTO 10
```

Digitate RUN e premete RETURN.
In entrambi gli esempi visti, la figura è rimasta ferma in un'area sullo schermo. La funzione TAB vi aiuta quando volete muovere gli oggetti dal bordo sinistro. (La funzione TAB è spiegata in dettaglio nell'enciclopedia BASIC). Il seguente programma mostra un serpente che striscia sullo schermo.

Ricordate che lo SHIFT e il tasto seguente vanno premuti assieme.

```
5 FOR A=0 TO 30
10 PRINT " SHIFT CLR "
20 PRINT TAB (A) " SHIFT U SHIFT I SHIFT U SHIFT I "
30 PRINT TAB (A) " SHIFT K SHIFT J SHIFT K SHIFT J SHIFT Q"
40 FOR L=1 TO 100: NEXT L
50 PRINT " SHIFT CLR "
60 PRINT TAB (A+1) " SHIFT I SHIFT U SHIFT I SHIFT U SHIFT W"
70 PRINT TAB (A+1) " SHIFT J SHIFT K SHIFT J SHIFT K "
80 FOR L=1 TO 100: NEXT L
90 NEXT A
```

Usando i caratteri come la pallina (SHIFT Q), potete realizzare dei video giochi. Per muovere la pallina basta cancellare la pallina e metterla nella nuova posizione, come in questo programma:

```
10 PRINT " SHIFT CLR "
20 PRINT " SPACE SHIFT Q██";
30 FOR L=1 TO 50: NEXT L
40 GOTO 20
```

cursore a sinistra

Quando lanciate (RUN) un programma, ricordatevi di premere RUN/STOP quando volete fermarlo.

CONTROLLO DEL COLORE

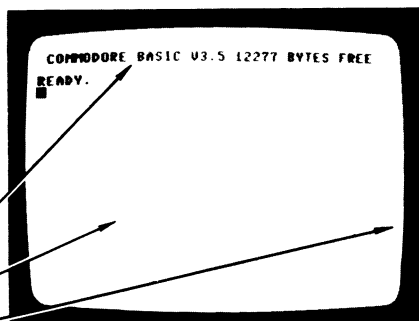
Diversi colori possono essere messi in ogni parte dello schermo. Lo sfondo può avere un colore, il bordo un altro, e ogni carattere un suo proprio colore. Già sapete come cambiare i colori usando la tastiera. Potete cambiare il colore del resto dello schermo usando il comando COLOR del BASIC.

Per esempio, potete far diventare il bordo dello schermo rosso digitando il comando COLOR 4,3 seguito da RETURN. Il numero 4 nel comando COLOR indica che stiamo operando sul bordo, mentre il numero 3 rappresenta il colore rosso (come sulla tastiera).

Provate a digitare COLOR 0,7 seguito da RETURN. Lo sfondo diventerà blu. Il numero 0 indica che stiamo operando sullo sfondo, mentre il numero 7 è il colore blu (come sulla tastiera).

Il primo numero dopo la parola COLOR specifica l'area di schermo che volete cambiare. La seguente tabella elenca il numero di controllo per ogni area; saprete di più sulle aree 2 e 3 leggendo la grafica multicolor più avanti in questo capitolo.

Numeri delle Aree di Schermo

AREA #	NOME DELL'AREA
0	Sfondo
1	Caratteri
2	Multi-color 1
3	Multi-color 2
4	Bordo

Il secondo numero dopo COLOR seleziona il colore che volete per l'area di schermo specificata. I numeri del colore corrispondono a quelli sulla tastiera.

Numeri di Colore

#	COLORE	#	COLORE
1	NERO	9	ARANCIO
2	BIANCO	10	CASTANO
3	ROSSO	11	GIALLO VERDE
4	CIANO	12	ROSA
5	VIOLA	13	BLU VERDE
6	VERDE	14	BLU CHIARO
7	BLU	15	BLU SCURO
8	GIALLO	16	VERDE CHIARO

Ogni colore ha anche un livello di luminosità regolabile, chiamato "luminanza". Potete aggiungere un numero da 0 (più scuro) a 7 (più chiaro) dopo il numero di colore per variare il colore. Digitate COLOR 4,3,0 e premete RETURN Il bordo diventa rosso scuro.

In breve, il comando COLOR funziona così:
COLOR area, colore, luminanza
Questo piccolo programma vi farà vedere tutti i colori del Commodore 16:
Prima di tutto digitate NEW seguito da RETURN

```
10 COLOR 0, 7, 7
20 FOR M=0 TO 7
30 FOR N=1 TO 2
40 FOR L=1 TO 16
50 PRINT " CTRL RVS ON" ; (premete questi tasti insieme)
60 READ A
70 COLOR 1, A, M
80 PRINT "SPACE SPACE"; (lasciare 2 spazi fra le virgolette)
90 NEXT L
100 PRINT
110 RESTORE
120 NEXT N,M
130 COLOR 1, 2, 4
200 DATA 7,14,4,13,6,16,11,8,10,9,3,12,5,15,2,1
```

Quando lanciate questo programma, lo sfondo dello schermo diventa blu chiaro, e l'intera gamma dei colori del Commodore 16 è mostrata con ogni livello di luminanza. Noterete che il nero è uguale a tutti i livelli di luminanza.

Nota: Come la maggior parte dei comandi grafici BASIC visti in questo capitolo, COLOR può essere usato sia in modo programma che in modo diretto. La distinzione non è importante nello spiegare COLOR, o qualunque altro comando grafico, dato che tale distinzione è basata sulla frequenza d'uso in modo diretto o in modo programma.

IL COMANDO GRAPHIC

I grafici che avete visto usare finora usano solo la tastiera, senza avvantaggiarsi delle capacità grafiche del computer. Il linguaggio BASIC del Commodore 16 contiene comandi per disegnare delle figure nei programmi. Per usare i comandi grafici del vostro computer, dovete selezionare un nuovo modo, il modo GRAPHIC.
Il modo GRAPHIC può essere considerato il modo per disegnare, dato che tutti i comandi per disegnare vanno attivati così. Non potete usare questi comandi fino a quando non specificate esattamente quale modo grafico volete.
In generale il comando GRAPHIC si usa in questo modo: GRAPHIC + tipo di grafica + (eventualmente) cancellazione

GRAPHIC

Modo	**Effetto**
0	Testo
1	Grafica ad alta risoluzione
2	Grafica ad alta risoluzione + testo
3	Grafica multi-colore
4	Grafica multi-colore + testo

Cancellazione	**Effetto**
0	Non cancella lo schermo
1	Cancella lo schermo

Per commutare dalla grafica normale (o modo testo) all'alta risoluzione, digitate il comando GRAPHIC 2, 1 e premete RETURN Lo schermo si cancella e il cursore riappare vicino al fondo dello schermo. Lo schermo del Commodore 16 è diviso in due sezioni:
La parte alta per i grafici e le ultime 5 linee in basso per il testo. Se non volete le 5 linee per il testo potete usare il comando GRAPHIC 1, 1, ma non potrete vedere i comandi che digiterete. Potete entrare e uscire dal modo grafico usando il comando GRAPHIC. Il comando GRAPHIC 0 fa tornare lo schermo in modo testo, mentre GRAPHIC 2 torna all'alta risoluzione senza cancellare lo schermo. Aggiungere 1 dopo il comando per cancellare lo schermo.
C'è un altro modo per cancellare lo schermo in alta risoluzione. Il comando SCNCLR cancella lo schermo senza cambiare il modo grafico. Quando usate la grafica ad alta risoluzione, il computer si riserva 10 K di memoria per lo schermo ad alta risoluzione. Questa memoria è tolta dall'area per i programmi BASIC. Quando avete finito di usare la grafica potete riottenere questa memoria usando il comando GRAPHIC CLR.

GRAFICA AD ALTA RISOLUZIONE

Il Commodore 16 ha una capacità di 25 linee di 40 caratteri ciascuna, o 1000 caratteri totali sullo schermo. Ogni carattere è formato di singoli punti con otto linee di punti di otto punti ciascuna per fare un intero carattere. Lo schermo ha un totale di 320 punti per riga e 200 righe di punti, o 64000 punti in totale. La grafica ad alta risoluzione del Commodore 16 vi dà il controllo di ogni singolo punto.
Usando la grafica normale avete un controllo limitato sui singoli punti. Disegnare una serie di bande colorate o una carta da gioco è un esercizio divertente, ma la vostra creazione è limitata all'uso dei caratteri (lettere, simboli grafici, etc.) sulla tastiera del computer; certo in questo modo potete creare prima o poi, tutte le figure che volete, ma questo è niente in confronto a ciò che potete fare controllando uno per uno tutti i puntini luminosi. Potete ancora creare tutte le figure. La grafica ad alta risoluzione del Commodore 16 vi permette di farlo. La risoluzione è la precisione del disegno; con la grafica ad alta risoluzione potete disegnare e cancellare punti, linee, cerchi e altre figure. L'alta risoluzione grafica ha però una limitazione. Il computer può usare solo due colori in ogni posizione di carattere 8x8. Cioè ogni spazio 8x8 sullo schermo (quello che sarebbe occupato da un singolo carattere) è limitato a due colori (sfondo e primo piano per quel quadrato). Potete usare differenti colori per ogni differente posizione di carattere, ma solo due colori entro una posizione. Un altro modo grafico che sarà spiegato più avanti in questa sezione, il modo multi-colore, vi permette di usare fino a 4 colori differenti per ogni posizione di carattere, a scapito della risoluzione disponibile nel modo ad alta risoluzione.
Ecco un programma che utilizza alcune delle caratteristiche grafiche ad alta risoluzione del vostro computer ed in particolare il comando GRAPHIC.
Cancellate la memoria del computer col comando NEW in modo da esser sicuri che non ci siano linee di programma "dimenticate" e digitate:

```
10 COLOR 0,1
20 GRAPHIC 1,1
30 FOR L=2 TO 16
40 COLOR 1,L,2
50 DRAW 1,0,L*12 TO 319,L*12
60 DRAW 1,1*18,0 TO L*18,199
70 NEXT L
80 FOR L = 1 TO 5000: NEXT
90 COLOR 1,2,3
100 GRAPHIC 0
```

Nota che il colore cambia vicino alle intersezioni. Questo è dovuto alla limitazione dei grafici ad alta risoluzione, con troppi colori da unire insieme.

Punti, Linee, Labels

Scrivi il comando GRAPHIC 2,1: DRAW 1,0,0 e premi RETURN Guarda attentamente il lato in alto a sinistra dello schermo. Il tuo COMMODORE 16 vi ha disegnato un punto nero. Il comando DRAW può essere usato per disegnare un singolo punto, una linea, una figura in qualsiasi zona dello schermo.

Alcune forme del comando DRAW sono:

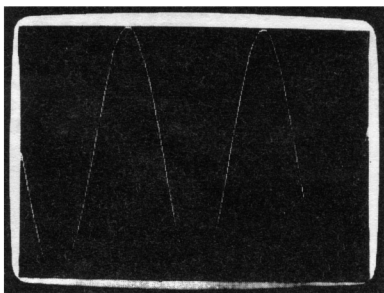
COMANDO	RISULTATI
DRAW colore, linea, colonna	PUNTO
DRAW colore, linea, colonna TO linea, colonna	LINEA
DRAW colore TO linea, colonna	LINEA DISEGNATA DALL'ULTIMO PUNTO

Il colore è 0 per lo sfondo (il punto non si vede), 1 per il primo # piano (il punto viene visualizzato).
Qualsiasi cosa disegnata col colore dello sfondo (0) ha per effetto la cancellazione del colore di primo # piano in quel punto.
Nel comando DRAW, il primo numero è sia 1 (disegna un punto) sia 0 (cancella un punto). I due numeri seguenti sono per la posizione di linea e colonna del punto. Così se tu vuoi disegnare un punto nella linea 17, colonna 20, devi scrivere DRAW 1, 17, 20. Per cancellare il medesimo punto devi scrivere DRAW 0, 17, 20.

Il comando DRAW può anche disegnare una linea tra due punti qualsiasi. Basta che tu aggiunga la parola TO e le coordinate dell'altro punto come in questo caso: DRAW 1, 1, 1 TO 100, 100. Questo disegna una linea dal punto di coordinate 1,1 al punto 100,100. Puoi cancellare questa linea scrivendo il medesimo comando, ma sostituendo uno 0 all'1 immediatamente seguente il comando DRAW. Se sei abituato a disegnare grafici in matematica sarai un poco confuso le prime volte. Il sistema di coordinate nel tuo Commodore 16 è differente da quello che hai sinora usato. Normalmente il punto di coordinate 0,0 è o al centro dello schermo o nell'angolo in basso a sinistra, ma nel tuo computer è nel lato in alto a sinistra. Ti ci abituerai con la pratica.

Una volta che hai disegnato un punto o una linea sullo schermo, ne puoi disegnare un'altra in un qualsiasi altro punto partendo dalla precedente. Per esempio: DRAW 1 TO 150,150. Questo comando disegna una linea dall'ultimo punto apparso sullo schermo al punto di coordinate 150,150. Se il tuo programma usa molti comandi DRAW, puoi piazzare il primo punto in una posizione dello schermo usando il comando LOCATE, come in LOCATE 100,100 per tornare immediatamente alla posizione iniziale.

Questo programma disegna una curva basata sulla funzione seno.

```
NEW 10 COLOR 0,1
10 COLOR 0,1
20 COLOR 1,2
30 GRAPHIC 1,1
40 LOCATE 0,100
50 FOR X= 1 TO 319
60 Y=INT (100+99*SIN(X/20))
70 DRAW 1 TO X,Y
80 NEXT X
90 FOR L=1 TO 5000
100 NEXT L
110 GRAPHIC 0
```

Non scrivere NEW dopo aver dato il RUN a questo programma. Per far funzionare diversamente il programma, cambia la linea 70 così:

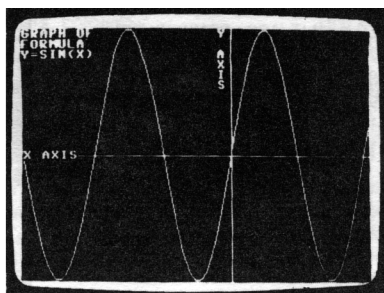
```
70 DRAW 1,X,Y
```

Questo programma disegna la medesima curva usando punti invece di linee.

Il Comando Char

I grafici sono più facili da capire e più utili se ad ognuno fai corrispondere una determinata label, ciè dai ad ognuno di essi una "etichetta" che ti permetta di distinguerli. Per esempio il comando CHAR 1,0,5, "HELLO" inserisce la parola HELLO nella quinta linea sotto l'angolo in alto a sinistra dello schermo. Il primo numero dopo l'istruzione CHAR è o 1 o 0, a seconda che tu voglia disegnare o cancellare la scritta. I due numeri seguenti rappresentano la colonna e la linea dove il testo appare.
Lascia gli ultimi due programmi nella memoria del computer: non scrivere NEW.

Aggiungi queste linee:

```
81 CHAR 1,0,0,"GRAFICO DELLA": CHAR 1,0,1, "FUNZIONE"
82 CHAR 1,0,2, "Y = SIN(X)"  .
83 DRAW 1,0,100 TO 319,100,189,0 TO 189,199
84 CHAR 1,0,12, "ASSE X": CHAR 1,22,3,"Y"
85 CHAR 1,22,2, "A": CHAR 1,22,3, "S"
86 CHAR 1,22,4, "S": CHAR 1,22,5, "E"
```

QUADRATI, CERCHI, POLIGONI, E COLORAZIONI

Disegnare Rettangoli

Usando il comando DRAW puoi disegnare figure utilizzando molti punti o linee. Per disegnare un quadrato, puoi usare il comando DRAW 1,0,0 TO 100,0 TO 100,100 TO 0,100 TO 0,0 per congiungere quattro punti con quattro linee (indicando solamente le coordinate dei quattro vertici del quadrato) oppure puoi utilizzare il comando BOX.

Il comando BOX

Uno dei comandi grafici del tuo Commodore 16 rende più facile disegnare quadrati ed altre figure rettangolari. Il comando BOX ti permette di indicare solamente le coordinate di due vertici opposti. Per disegnare il medesimo quadrato come nell'esempio di prima scrivi: BOX 1,0,0,100,100. Il numero 1 significa ancora che devi disegnare e non cancellare il quadrato; gli altri quattro numeri sono le coordinate del vertice in alto a sinistra (0,0) e del vertice opposto in basso a destra (100,100).

Il comando BOX può formare qualsiasi rettangolo; basta cambiare le coordinate dei vertici. Puoi anche ruotare il quadrato specificando un angolo in gradi dopo l'ultima coordinata, come in questo caso: BOX 1,50,50,100,100,45. Il quadrato è ora ruotato di 45 gradi in senso orario così da sembrare un rombo.
Se non ti accontenti del semplice contorno del rettangolo, puoi colorare il suo interno. Apparirà per esempio un rettangolo dipinto al centro dello schermo digitando il comando BOX 1,100,50,220,150,,1. Nota che hai bisogno di una virgola invece del valore dell'angolo, nel caso che tu non voglia che la figura venga ruotata. Il tuo computer interpreta la virgola come valore di "DEFAULT", vale a dire che interpreta la mancanza di un numero come una ben precisa istruzione.
Se non includi la virgola l'1 alla fine della linea è letto come l'angolo di rotazione della figura.

Alcune tipiche forme del comando BOX

COMANDO	RISULTATO
BOX 1, linea 1, colonna 1, linea 2, colonna 2	Semplice contorno
BOX 1, linea 1, colonna 1, linea 2, colonna 2, angolo, 1	Figura colorata
BOX 0, linea 1, colonna 1, linea 2, colonna 2, angolo, 0	Cancella una parte dello schema

Linea 1 colonna 1 sono le coordinate del vertice in alto a sinistra; linea 2 colonna 2 sono le coordinate del vertice in basso a destra

Ecco un programma che utilizza il comando BOX (in linea 60)

```
10 COLOR 0,1
20 COLOR 1,2
30 GRAPHIC 2,1
40 A = RND(1)*20 + 10
50 FOR L = 0 TO 359 STEP A
60 BOX 1,100,30,220,130,L
70 NEXT L
80 FOR L = 1 TO 2000: NEXT L
90 GRAPHIC 0,1
```

Per fermare questo programma devi premere RUN STOP

Disegnare Cerchi

Il tuo Commodore 16 possiede anche comandi per disegnare cerchi. Come il comando BOX, puoi variare la forma del cerchio (per formare un ovale o una ellisse), e puoi ruotare l'ovale. Puoi anche disegnare una parte di cerchio (cioè un arco). Le tipiche forme del comando CIRCLE sono:

COMANDO	RISULTATO
CIRCLE 1, colonna centrale, linea centrale, raggio del centro	OVALE
CIRCLE 1,y centrale, x del centro, lunghezza, altezza	CIRCONFERENZA
CIRCLE 1,y centrale, x del centro, lung., alt., inizio, fine	ARCO
CIRCLE 1,y centrale, x del centro, lung., alt.,,,,angolo	OVALE RUOTATO
CIRCLE 1,y,x,lung., alt,,,,angolo del punto	POLIGONO

Il seguente comando disegna un cerchio nello schermo: CIRCLE 1,160,100,50. Cioè dice al computer di disegnare un cerchio con centro in 160,100 e di raggio 50. In realtà esso però produce un ovale, poiché a volte i punti dello schermo sono più alti che larghi. Per cambiare questa figura in una vera circonferenza devi aggiungere un numero separato per dire che l'altezza è differente dalla larghezza come nel caso: CIRCLE 1,160,100,50,42.

Disegnare Poligoni

Il tuo Commodore 16 può anche disegnare un quadrato, un triangolo, un qualsiasi altro poligono usando il comando CIRCLE. Basta che tu dica al computer quanti gradi ci devono essere tra i punti della circonferenza, come in: CIRCLE 1,160,100,50,42,,,,120. Questo comando disegna un triangolo equilatero poiché l'angolo al centro fra due vertici consecutivi della figura è 120 gradi (Ricorda che omettere i valori dei numeri inserendo al loro posto le virgole in un comando grafico causa la lettura del valore limite standard del numero mancante). Una semplice formula per trovare l'angolo di un poligono con N lati è: 360/N.

Ecco un breve programma per disegnare poligoni:

```
10 GRAPHIC 2,1
20 INPUT "QUANTI LATI";A
30 IF A < 2 or A > 100 THEN PRINT "NON ESSERE RIDICOLO": GOTO 20
40 CIRCLE 1,160,80,40,33,,,,360/A
50 GOTO 20
```

Puoi scegliere di disegnare solo un arco invece di una intera circonferenza. Il comando CIRCLE 1,160,100,50,42,90,180 mostra solo la sezione più bassa di una circonferenza.
Per ruotare un ovale, aggiungi l'angolo di rotazione in senso orario dopo il comando, come in questo esempio: CIRCLE 1,160,100,100,20,,,30.

Ecco un programma che usa il comando CIRCLE per un effetto interessante. Ricordati di pulire la memoria del tuo computer scrivendo NEW se il programma precedente è ancora in memoria.

```
10 COLOR 0,1
20 COLOR 1,2
30 GRAPHIC 1,1
40 A=RND(1)*2010
50 FOR L = 0 TO 359 STEP A
60 CIRCLE 1,160,100,80,40,,,L
70 NEXT L
80 FOR L = 1 TO 2000: NEXT L
90 GRAPHIC 0,1
```

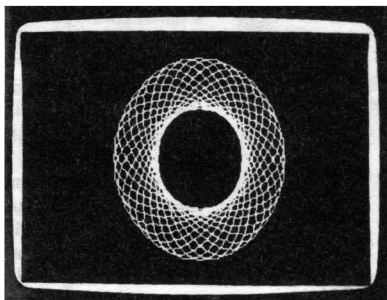
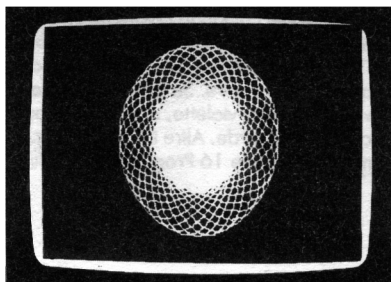



figura: cosa appare sullo schermo

Il Comando Paint

Il comando "PAINT", riempie una qualsiasi area delimitata da un segno (una figura geometrica per esempio), o, in mancanza di questo, un'intera porzione dello schermo. Anche il comando BOX ha un'opzione di 'riempimento' per colorare quadrati e rettangoli, ma PAINT può servire per colorare qualunque area per irregolari che siano i suoi contorni.
Per vedere come funziona aggiungete al programma precedente queste istruzioni:

```
75 PAINT 1,160,100
```

Figura che appare sullo schermo

GRAFICA MULTICOLORE

La grafica ad alta risoluzione del Commodore 16 vi permette di controllare ogni singolo punto luminoso (pixel) dello schermo; purtroppo come avete visto esistono dei problemi con i colori: la gran parte dei programmi ad alta risoluzione ne possono usare solo due.
Per aggiungerne altri il vostro computer è dotato di uno speciale "modo" intermedio chiamato "grafica multicolore". In questo "modo" la risoluzione però è dimezzata: i punti minimi di controllo sono 160x200, la metà dei punti luminosi controllati in alta risoluzione, si tratta però di un compromesso necessario.

Per vedere come lavorare in grafica multicolore dobbiamo ricordarci i comandi "graphic" già visti all'inizio del capitolo: con GRAPHIC 3 abbiamo uno schermo multicolore senza testo, mentre con GRAPHIC 4 possiamo disporre di cinque righe di testo.
Ora pensiamo ai comandi di colore: c'erano due aree che non avevamo ancora usato: la 2 e la 3. Queste aree permettono di usare due colori extra oltre a quelli del testo, colori che non interferiscono gli uni con gli altri.
Con questo programma utilizziamo la grafica multicolore per ottenere un curioso effetto da insegna al neon:

```
NEW
10 COLOR 0,1
20 GRAPHIC 3,1
30 COLOR 3,1
40 TRAP 200
50 DRAW 3,10,10 TO 10,100: DRAW 3,10,55 TO 30,55
60 DRAW 3,30,10 TO 30,100: DRAW 3,50,10 TO 80,10
70 DRAW 3,65,10 TO 65,100: DRAW 3,50,100 TO 80,100
80 FOR L = 0 TO 7
90 COLOR 3,2,L
100 FOR M=1 TO 100: NEXT M
110 NEXT L
120 COLOR 3,1
130 FOR M=1 TO 100: NEXT M
140 GOTO 80
200 GRAPHIC 0: COLOR 1,2,7
```

L'area 3 del comando colore ha invece una speciale capacità. Una volta che avete disegnato sullo schermo usando questo comando, basterà dare l'istruzione relativa ad un cambiamento di colore per vedere cambiare tutte le aree di quel colore. Se disegnate delle aree con l'istruzione COLOR 3,5 queste verranno color violetto, dopodiché basterà un comando COLOR 3,6 per colorarle di verde. Altre informazioni a proposito della grafica sono contenute nel Commodore 16 Programmer's Reference Guide.

CAPITOLO 7

SUONI E MUSICA

- Introduzione
- Comandi di volume
- Comandi di suono
- Creazione di effetti sonori
- Come fare un po' di musica
- La macchina della Musica

INTRODUZIONE

Ecco un breve programma da fare con il vostro Commodore 16. Una volta che lo avete digitato e fatto girare, vedrete apparire sullo schermo un punto interrogativo. Scrivete un numero qualsiasi da 0 a 1015 come risposta e premete RETURN. Per fermare il programma rispondete col valore 0 o premete il solito RUN STOP.

```
10 VOL 8
20 DO
30 INPUT X
35 IF X > 1015 OR X < 0 THEN PRINT "SI ACCETTANO SOLO NUMERI DA
0 A 1015!": GOTO 30
40 SOUND 1, X, 10
50 LOOP UNTIL X=0
```

Questo programma fa emettere al vostro Commodore 16 un suono della frequenza da voi decisa.
Ecco invece come far emettere una singola nota: per prima cosa digitate VOL 8 e premete RETURN, poi scrivete SOUND 1, 266, 60 e premete RETURN. Per circa un secondo ascolterete una nota. Non è molto ma con un po' di buona volontà la si può considerare una "sinfonia incompiuta". Se non sentite niente controllate il volume del vostro televisore o del monitor e riprovate.
VOL e SOUND sono gli unici due comandi che dovete conoscere per fare della musica con il vostro Commodore 16; si tratta come vedremo di comandi facili da capire e da usare.

Il Comando di Volume

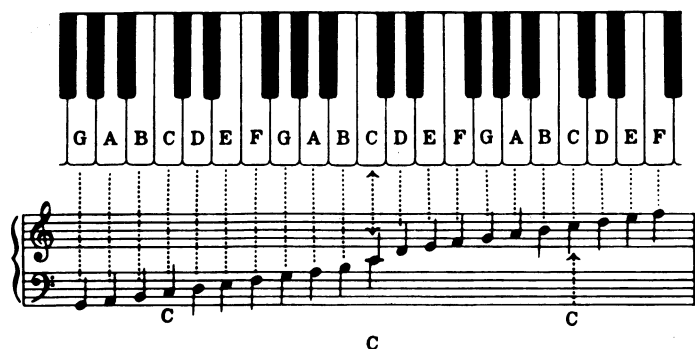
L'istruzione VOL controlla appunto il VOLume del suono emesso dal computer. Il numero che segue l'istruzione (ad esempio "8" in VOL 8) indica il livello del volume. Tutto sommato questo comando funziona un po' come una manopola del volume in un impianto stereo o in un registratore: con VOL 0 non si sentirà niente, mentre con VOL 8 il suono è al massimo volume possibile.

Il Comando di Suono

Il comando SOUND è un po' più complesso: è seguito infatti da tre numeri che indicano esattamente al computer il suono che volete ottenere: la struttura del comando è la seguente:

SOUND voce, valore della nota, durata

Esempio SOUND 1, 266, 60

Il primo numero indica la "voce" e può essere 1, 2 o 3. Il Commodore 16 ha due "voci" selezionabili con 1 e 2 ma mentre la prima produce solo toni, la seconda può produrre toni (con l'istruzione 2) o rumori (con l'istruzione 3). L'istruzione SOUND 3, può servire per ottenere effetti come tuoni, pioggia etc.
Il secondo numero indica la frequenza della nota. Come abbiamo visto può trattarsi di qualsiasi numero compreso tra 0 e 1015. Indica al vostro computer quanto sordo o acuto deve essere il suono da emettere: più il numero è alto più il suono è acuto: i valori più alti (vicini al 1015) sono addirittura ultrasuoni, frequenze talmente alte da non potere essere uditi dall'orecchio umano.
Nell'area 600-940 la voce due produce (con l'istruzione 3) un interessante rumore "bianco" ottimo per ottenere effetti speciali.
Attenzione però a non confondere la frequenza vera (quella misurata in fisica con gli Hertz) con il numero dato: si tratta di scale di valori proporzionali ma non coincidenti.
Per esempio ecco una tabella con i valori dell'istruzione corrispondenti alle sette note e la relativa frequenza in Hertz:

NOTA	LA	SI	DO	RE	MI	FA	SOL
VALORE	770	798	810	834	854	864	881
HERTZ	440,4	494,8	522,7	588,7	658	699	782,2

Ora proviamo ad ascoltare le capacità sonore del Commodore 16 con questo breve programma che, grazie ad un loop, ci darà tutte le frequenze multiple di 5 fino a 1020.

```
NEW
10 VOL 7              <------------   (livello di volume)
20 X = 0
30 DO
40 SOUND 1, X, 5 <------------   (Istruzione per l'esecuzione)
50 x = X + 5
60 LOOP UNTIL X = 1020
70 VOL 0              <------------   (Azzeramento volume)
80 END
```

E il terzo numero? Controlla la durata del suono, cioè quanto a lungo il computer "terrà" questa nota. Può trattarsi di qualsiasi numero da 0 a 65535.
Il numero regola infatti un contatempo: ogni unità corrisponde ad un sessantesimo di secondo. Nell'esempio precedente quindi ogni suono viene eseguito per 5/60 di secondo: se avessimo un istruzione:

```
SOUND 1, 800, 65535
```

Un suono della frequenza indicata con 800 verrebbe eseguito ininterrottamente per oltre mille secondi (più di sedici minuti): con un valore zero il suono non verrà neppure emesso.

Effetti sonori

Possono essere ottenuti usando toni musicali o rumori combinati con semplici istruzioni Basic. Il loop FOR... NEXT... STEP per esempio può essere usato in maniera molto creativa. Nell'esempio precedente un ciclo di questo tipo faceva variare il valore di x di cinque unità alla volta. In questo modo il programma faceva eseguire successivamente suoni di frequenze diverse (la variabile indicava infatti la frequenza del suono). La stessa cosa accade con il programma seguente, dove il computer eseguirà l'uno dopo l'altro per la durata di un solo sessantesimo di secondo suoni di tutte le frequenze multiple di 25 dal valore 1000 al 700 si tratterà quindi di una 'scala' musicale tendente verso suoni più sordi.

```
10 VOL 8
20 FOR S=1000 TO 700 STEP -25 (ecco il loop, notare gli "step" al contrario)
30 SOUND 1, S, 1
40 NEXT S
```

Premiamo RUN e RETURN e ascoltiamo l'effetto: come è successo? La chiave è la linea 20 che seleziona un intervallo tra i valori 1000 e 700 istruendo il computer ad emettere frequenze decrescenti ma multiple di 25. Il computer quindi eseguirà l'istruzione 30 emettendo la frequenza 1000, per un attimo con la voce uno, poi leggerà l'istruzione 40 che lo fa tornare a 20, con un veloce calcolo troverà che il numero seguente è 975 ed eseguirà un suono con frequenza 975 e così via. Esperimenti con loop simili associati alla durata, al volume e naturalmente alla frequenza del suono permettono di creare interessanti effetti.

Creazione di Rumori

Con il valore 3, selezioniamo un rumore. Vediamo un programma che utilizza questa opzione per ricreare il rumore del vento in una tempesta:

```
10 VOL 2
20 R= INT (RND(0)*10)+1     (seleziona a caso numeri da 1 a 0)
30 FOR X=1 TO R
40 SOUND 3, 600+30*X, 10
50 NEXT X
60 FOR X = R TO 1 STEP -1
70 SOUND 3, 600+30*X,10
80 NEXT X
90 T=INT (RND(0)*100)+30
100 SOUND 3,600,T
110 GOTO 20
```

Le linee 30 e 60 creano dei loop di FOR... NEXT che variano in senso crescente e decrescente il valore delle frequenza nelle istruzioni 40 e 70. Questa variazione è importante per rendere i differenti rumori del vento. Le linee 90 e 100 selezionano invece un numero casuale per ricreare i silenzi tra un soffio e l'altro tipici del vento in tempesta. Il programma ha due comandi di SOUND per creare un "controcanto" e una pluralità di voci anch'essa conforme alla natura delle tempeste.
Può sembrare un po' complicato, e in effetti è necessaria una certa esperienza di programmazione per inventarne di analoghi, in realtà basta fare degli esperimenti, creare un programma e vedere cosa succede. Trarre effetti sonori dai rumori è una vera e propria sfida, un tentativo di catturare gli elementi costitutivi e caratteristici del fenomeno che volete riprodurre.

Come Fare un po' di Musica

In ogni caso non preoccupatevi se non capite sempre tutto di questi programmi, iniziate a digitarli e state a guardare cosa succede:.
Questo programma per esempio fa diventare il vostro Commodore 16 un pianoforte, trasformando i numeri da 1 a 8 in creatori di note.

```
5 SCNCLR
10 FOR X=1 TO 8 : READ N(X) : NEXT X
20 VOL 7
30 DO
40 GET A$:IF A$= "" THEN 40
50 A=ASC(A$):IF A<49 OR A > 56 THEN 90
60 N=A-48
70 SOUND 1,N(N),5
80 COLOR 0,N,3
90 LOOP UNTIL A=32
100 VOL 0: COLOR 4,2,7
110 DATA 169, 262, 345, 383, 453, 516, 571, 596
```

Premendo i tasti numerici da 1 a 8 si suoneranno delle note musicali, e il bordo dello schermo cambierà colore a seconda delle note. Per fermare il programma basterà premere la barra spaziatrice.
Ora che il vostro Commodore 16 è diventato un pianoforte potreste voler suonare qualche motivetto popolare: ecco le istruzioni per suonare una canzoncina:

Ma perché digitare tutti questi numeri ogni volta? La musica potrebbe essere eseguita da programma. Ecco un programma che suona leggendo i numeri contenuti nelle istruzioni di DATA come valori da assegnare a due variabili, in questo caso X e Y che comandano rispettivamente la frequenza e la lunghezza del suono.
Per esempio nella linea 100 il numero 169 indica una certa frequenza che verrà eseguita per 45 sessantesimi di secondo, e così via.

```
10 VOL 8
20 DO
30 READ X,Y
40 SOUND 1,X,Y
45 FOR D=1 TO Y*16 +30:NEXT
50 LOOP UNTIL X= 0   (per creare brevi spazi fa le note)
60 END
100 DATA 169, 45, 169, 45, 169, 30
```

```
110 DATA 262, 15, 345, 45, 345, 30
120 DATA 262, 15, 345, 30, 383, 15
130 DATA 453, 60, 596, 45, 453, 45
140 DATA 345, 45, 169, 45, 453, 30
150 DATA 383, 15, 345, 30, 262, 15
160 DATA 169, 60
200 DATA 0, 0
```

LA MACCHINA DELLA MUSICA

Quest'ultimo programma è un po' più lungo: serve a trasformare il Commodore 16 in una "macchina per la musica" si chiama infatti "la macchina della musica". Basterà premere un tasto numerico da 1 a 9 per ascoltare la nota desiderata e vederla sul rigo giusto del pentagramma disegnato sullo schermo.

```
5 GOSUB 1000
6 FOR X=1 TO 9 : READ N(X): NEXT X
8 CHAR 1,8,1, "**LA MACCHINA DELLA MUSICA**"
10 VOL 7
20 DO
30 GET A$ : IF A$ = "" THEN 30
35 A=ASC(A$) : IF A<49 ORA >57 THEN 50
36 N=A-48
40 SOUND1,N(N),4
45 GSHAPE N$,150,8*(6+(9-N)),4
46 FORZ=1 TO50:NEXTZ
47 GSHAPE N$, 150,8 * (6+(9-N)),4
50 LOOP UNTIL A=32
55 VOL 0 : GRAPHIC 0 : SCNCLR
60 END
100 DATA 345, 383, 453, 516, 571, 596, 643, 685, 704
1000 GRAPHIC 1,1
1010 FOR Y = 60 TO 124 STEP 16
1020 DRAW 1,100, Y TO 200, Y
1030 NEXT Y
1040 A$= "FEDCBAGFE"
1050 FOR X= 1 TO 9: C=13
1060 IF INT (X/2) = X/2 THEN C=14
1070 CHAR 1,C,X+6,MID$ (A$,X,1),0
1075 CHAR 1, C+10, X+6, RIGHT$ (STR$ (10-X) , 1)
1080 NEXT X
1090 FOR X=1 TO 8 : FDR Y = 11 TO 16 : DRAW 1,X,Y: NEXT Y, X
1100 Y=1: X=8 : DRAW 1,8,16, TO X, Y
1110 SSHAPE N$ 1,1,8,16
1120 GSHAPE N$ 1,1,4
1130 RETURN
```

Come potete vedere la musica e il suono possono essere utilizzati all'interno di programmi più vasti o essere l'oggetto stesso del programma. Gli esempi forniti in questo capitolo vi danno solo un'idea delle capacità musicali del Commodore 16. Sperimentando altri suoni e rumori potrete creare il vostro capolavoro.

CONCLUSIONE

Abbiamo tentato di darvi qualche informazione su alcune delle cose che potrete fare con il vostro Commodore 16. Lo scopo di questo manuale è stato quello di darvi un'idea della programmazione, per spingervi a fare nuovi esperimenti, vedere cosa succede, imparare di più sul vostro computer e divertirvi. Ma c'è molto di più. Questo manuale è servito ad iniziarvi all'uso del Basic ma non è un manuale di Basic. La Enciclopedia Basic che segue vi darà una lista completa di TUTTI i comandi Basic del vostro Commodore 16 con spiegazioni ed esempi. In molti dei programmi inseriti negli ultimi due capitoli alcuni di voi possono non aver compreso esattamente quali comandi facessero eseguire al computer determinati compiti. L'Enciclopedia Basic potrà aiutarvi, ma se vorrete approfondire la programmazione potrete acquistare il Commodore 16 Programmer's Reference Guide o qualcuno dei numerosi testi sul linguaggio Basic e rinfrescare le vostre conoscenze seguendo le riviste specializzate.

BASIC 3.5

Enciclopedia

- Introduzione
- Comandi
- Istruzioni
- Funzioni
- Variabili e operatori
- Abbreviazioni

INTRODUZIONE

In questo manuale avete potuto vedere un vasto assortimento di programmi Basic; vi siete quindi fatti un'idea della programmazione e del suo particolare vocabolario. Questa Enciclopedia vi dà una lista completa delle regole e dei termini (sintassi) del linguaggio Basic 3.5 insieme ad una concisa descrizione di ciascuno di questi.
Esercitatevi pure con questi programmi ricordandovi che nessun programma, per sbagliato che sia, potrà mai danneggiare il vostro computer e che sperimentare è il miglior modo per imparare.
Lo scopo di questa Enciclopedia non è tanto quello di insegnare la programmazione in Basic, esistono molti libri in proposito, quanto quello di darvi brevi spiegazioni ed esempi sui comandi e le istruzioni del Basic 3.5.
Comandi e istruzioni sono elencati in ordine alfabetico in sezioni separate. I comandi sono usati principalmente in modo diretto mentre le istruzioni sono più frequentemente utilizzate all'interno di programmi. La maggior parte dei comandi però può essere inserita all'interno di un programma come istruzioni, e diverse istruzioni funzionano come comandi se inseriti in maniera diretta.
L'Enciclopedia Basic è divisa in quattro settori:
Comandi. I comandi utilizzati per lavorare con i programmi, correggerli, archiviarli, cancellarli etc.
Istruzioni. Le istruzioni Basic utilizzate all'interno dei programmi
Funzioni. Le funzioni di stampa, numeriche e di stringa
Variabili e operatori. I differenti tipi di variabili, i nomi "legali" di variabili e gli operatori aritmetici e logici.
Una spiegazione più completa del Basic 3.5 è contenuta nel Commodore 16 Programmer's Reference Guide disponibile in diversi negozi specializzati.

FORMATI STANDARD DEI COMANDI E DELLE ISTRUZIONI

I comandi e le istruzioni presentate in questa sezione dell'Enciclopedia seguono, per maggiore chiarezza, precisi e coerenti "formati", vale a dire strutture standard. Nella gran parte delle "voci" di questa Enciclopedia verranno dati degli esempi di questa struttura standard. Per esempio il comando LOAD da solo non ha nessun significato, bisogna specificare che cosa deve essere letto, e dove. Questi dati però non devono essere forniti in ordine casuale: ecco la struttura (il format) del comando LOAD

LOAD "Nomeprogramma", D0, U8

Parola chiave - argomento - altro argomento

All'inizio di ogni voce dell'Enciclopedia Basic che lo richiede è inserita la struttura del format. Per maggiore chiarezza sono stati adottati determinati simboli grafici eccone un elenco con relativa spiegazione.

Le parole chiave

(LOAD nel nostro esempio) rappresentano l'elemento centrale di una istruzione, sono essi ad indicare al computer che tipo di azione vi aspettate che egli compia, naturalmente nessuna variabile potrà mai avere lo stesso nome di una parola chiave.

Gli argomenti (o parametri)

Nella nostra enciclopedia sono scritti in minuscolo: si tratta della parte "variabile" del comando o dell'istruzione (siete voi a decidere il nome del programma che volete far leggere al computer). Completano l'istruzione o il comando indicando a che cosa esso è relativo. Vi possono essere anche altri argomenti opzionali come ad esempio il numero del drive (se ce ne sono diversi), la periferica che deve compiere una determinata azione etc.

Parentesi quadre []

Indicano gli argomenti (o parametri) opzionali che POSSONO O MENO essere indicati nell'istruzione a seconda delle vostre esigenze.

Parentesi ad angolo < >

Indicano DOVETE scegliere uno degli argomenti elencati

Barre verticali /

Separano i diversi elementi di una lista di argomenti. La lista è presente ogni qualvolta il numero degli argomenti inseribili è limitato. Quando una lista di argomenti è inserita TRA BARRE OBLIQUE ciò significa che sebbene il numero dei parametri sia limitato voi non siete obbligati ad inserirne uno.

Puntini ...

Un sere di tre punti indica che un determinato argomento può essere ripetuto più di una volta.

Virgolette " "

Racchiudono stringhe di caratteri, nomi di file e altre espressioni. Fanno parte integrante dell'istruzione e devono essere sempre messe, quando indicate.

Parentesi ( )

Lo stesso vale per le parentesi quando sono incluse in un format ciò significa che esse fanno parte integrante dell'istruzione relativa e devono essere digitate.

Variabili

Il termine indica qualsiasi nome corretto per variabili Basic come X, A$ o T%

Espressioni

Indica qualsiasi corretta espressione in Basic come A+B+2 o 5*(X+3)ZL

COMANDI BASIC

AUTO

AUTO [numero di linea]
Es. AUTO 10

Numera automaticamente le istruzioni, risparmiandovi il lavoro di digitarle. Ogni volta che avete inserito una linea di istruzioni con RETURN vedrete stampato sullo schermo il numero seguente col cursore nella posizione corretta per scrivere l'istruzione. Il numero che segue indica gli incrementi da effettuare. Per esempio AUTO 10 incrementerà il numero di linea di 10 (10 20 30 etc); AUTO 50 di 50 (50 100 150 etc).
Per interrompere la numerazione automatica basta digitare AUTO senza nessun numero.

BACKUP

BACKUP Ddrive TO D drive [, ON U periferica]
Esempio BACKUP D0 TO D1 'ON U9

Questo comando copia tutti i file presenti su un disco su di un altro disco. La copia è possibile anche senza aver inizializzato preventivamente con un comando di HEADER il dischetto "ricevente". Il comando di BACKUP infatti copia tutte le istruzioni presenti sul primo disco, inizializzazione e formattazione

compresa. Fare una copia di backup di tutti i propri dischi più importanti è una misura estremamente consigliabile.
Attenzione però, dato che il comando di Backup inizializza sempre il secondo disco, ne distrugge tutte le informazioni. Assicuratevi perciò che il disco su cui voi riversate le informazioni sia vuoto o che non contenga nessun dato da conservare. (Vedi anche COPY).
Questo comando può essere usato solo nei sistemi dotati di un doppio disk driver.

Esempi

BACKUP D0 TO D1	Copia tutti i file presenti nel disco inserito nel drive 0 sul disco inserito nel drive 1.
BACKUP D0 TO D1, ON U9	Copia tutti i file presenti nel disco inserito nel drive 0 sul disco inserito nel drive 1 all'interno dell'unità dischi numero 9

COLLECT

COLLECT [Ddrive] [, ON U unità]
Esempio COLLECT D0

Questo comando serve a liberare lo spazio assegnato a files mal chiusi cancellando il loro riferimento sulla DIRECTORY. I numeri che eventualmente seguono il comando servono a localizzare il disco.

CONT

CONT

Questo comando serve a riprendere l'esecuzione di un programma fermato con il tasto STOP, o con un'istruzione END o STOP da programma. Attenzione: CONT non funziona se, dopo aver fermato il programma avete eseguito qualche modifica. Se l'esecuzione si è fermata a causa di un errore il comando CONT produrrà un messaggio di errore CAN'T CONTINUE ERROR.

COPY

COPY [Ddrive,] "nome del file" TO [Ddrive,] "altro file" [, ON U unità]

Esempio

COPY D0, "CIAO" TO D1, "EHILA", ON U9

Questo comando esegue una copia di un file sullo stesso disco o su un altro, cambiando eventualmente il nome del file. Nel caso il nome rimanga invariato questo deve essere inserito due volte.

COPY "CIAO" TO "CIAO"

Esegue una copia cambiando nome sullo stesso disco

COPY D0, "CIAO" TO D1, "EHILA'", ON U9

Esegue sul disco presente nel drive 1 dell'unità dischi 9 col nome "EHILA'" una copia del file "CIAO" presente nella stessa unità sul disco inserito nel drive 0.

COPY D0 TO D1

Invece copierà sul disco inserito in D1, TUTTI i file presenti nel disco in D0

DELETE

DELETE [numero prima linea] [– numero ultima linea]
Esempio DELETE 10-50

Cancella una serie di linee presenti in un programma Basic, funziona solo in modo diretto.

Esempi

DELETE 75	Cancella la linea 75 di un programma
DELETE 10-50	Cancella tutte le linee dalla 10 alla 50 incluse
DELETE -50	Cancella tutte le linee dell'inizio del programma fino alla 50 inclusa
DELETE 75-	Cancella tutte le linee dalla 75 (inclusa) in poi fino alla fine del programma

DIRECTORY

DIRECTORY [Ddrive] [,Uunit] [,"nome del file"]
Esempio DIRECTORY D1,U9,"CIAO"

Questo comando fa scorrere sullo schermo la DIRECTORY (o catalogo) di un disco. Se si tratta di un elenco molto lungo premete contemporaneamente CTRL e S per fermarne lo scorrimento (per farlo ripartire basterà premere qualsiasi tasto) oppure C= per rallentarlo. Come si vede negli esempi questo comando permette di rintracciare file di cui non si conosce esattamente il nome.

Esempi

DIRECTORY	Elenca tutti i file presenti sul disco in uso
DIRECTORY D1, U9, "CIAO"	Elenca tutti i file chiamati "Ciao" presenti nel disco inserito nel drive 1 (eventualmente nell'unità dischi in U9)
DIRECTORY "AB*"	Elenca tutti i file che iniziano con AB su di un determinato disco
DIRECTORY "? CIAO"	Il punto interrogativo è un "Jolly", con questo comando vedrete elencati tutti i file con un carattere qualsiasi al posto del punto interrogativo e identici per il resto al nome inserito come "1 CIAO", "D CIAO" etc.

Nota: questo comando non può però essere usato per fare una copia su carta. A questo scopo è necessario inserire questi comandi (che cancellano tutto ciò che c'è in memoria al momento):

```
LOAD "$0",8
OPEN 4,4:CMD4:LIST
PRINT // 4:CLOSE4
```

DLOAD DLOAD "nome del file" [,Ddrive] [,Uunità]
Esempio DLOAD "CIAO",D0,U9

Questo comando carica in memoria un programma dal disco su cui sta lavorando se non ci sono specificazioni o su quello indicato

Esempi

DLOAD "CIAO"	Carica il programma CIAO
DLOAD (A$)	Carica il programma da un disco il cui nome è inserito nella variabile A$. Se A$ è vuoto avrete un messaggio di errore

NOTA 1   DLOAD può essere usato da programma per caricare e far girare un altro programma
NOTA 2   Se il programma è su CASSETTA usate LOAD

DSAVE DSAVE "nome del file" [,Ddrive] [,Uunità]

Questo comando salva, caricandolo sul disco attualmente in uso (se non vi sono specificazioni) o su quello indicato, il programma attualmente in memoria. Dovete dare il nome al programma.

Esempi

DSAVE "CIAO"	Salva il programma CIAO sul disco in uso
DSAVE "CIAO",D0,U9	Salva il programma "CIAO" sul disco presente nel drive 0 dell'unità corrispondente all'uscita 9

HEADER HEADER "nome del disco",Ddrive [,Inumero di identificazione][,ON Uunità]

Esempio HEADER "ILMIODISCO",I12,D0

Prima di utilizzare per la prima volta un nuovo dischetto dovete formattarlo con il comando HEADER. Dato che il comando HEADER ha l'effetto di cancellare tutto ciò che fino a quel momento era presente sul disco, potete usarlo anche solo con questo scopo. Formattare significa dividere la superfice magnetica del disco in settori chiamati "blocks" e per creare un indice dei programmi contenuti chiamato "Directory" o "Catalog". Come nome del disco può essere utilizzata qualsiasi stringa inferiore a 16 caratteri, mentre il numero di identificazione può averne solo due. Date a ogni disco un diverso numero di identificazione.
Senza il numero di identificazione la formattazione avviene più velocemente. Se il disco era già stato formattato potete non utilizzare il numero di identificazione, dato che l'effetto sarà quello di cancellare il directory senza formattare il disco.

HELP HELP

Il comando di HELP va utilizzato dopo che siete incappati in un errore. Una volta digitato HELP, vedrete apparire la riga di programma errata con la parte contenente l'errore in evidenza.

KEY KEY [tasto, stringa]
Esempio KEY 7, "GRAPHICO"+CHR$ (13) + "LIST" + CHR$ (13)

Sul vostro Commodore 16 ci sono otto tasti di funzione disponibili: quattro richiedono lo shift e quattro no. È possibile far corrispondere ad ognuna di queste una determinata serie di funzioni, anche complessa come nell'esempio ma non più lunga di 128 caratteri.
Da solo, il comando KEY produce una lista di tutte le funzioni assegnate ai singoli tasti.
Il comando KEY può anche essere inserito da programma per esempio:

```
10 KEY 2, "TESTING" + CHR$ (34): KEY 3, "NO"
```

oppure

```
10 FOR I=1 TO 8: KEY I, CHR$ (I + 132):NEXT
```

che assegna ai tasti di funzione gli stessi "compiti" assegnati sugli analoghi tasti del Commodore 64 e del VIC 20
Per tornare ai valori normali spegnete il computer o schiacciate RESET

LIST LIST [prima linea][-[ultima linea]]
Esempio LIST 10-50

Il comando LIST vi permette di visualizzare una o più linee di un programma Basic appena digitato o caricato in memoria. Quando è usato da solo tutto il programma verrà fatto scorrere. Per rallentarne lo scorrimento premete insieme C= per fermarlo temporaneamente CTRL e S (la pausa termina appena schiacciato un tasto qualsiasi) per fermarlo definitivamente schiacciate il tasto RUN STOP Se dopo la parola LIST inserite un solo numero questo mostrerà solo la linea richiesta, se ne inserite due collegati da una linetta, tutte quelle comprese nell'intervallo (comprese le due righe limite) scorreranno sul video. Se scrivete LIST seguito da un numero e da una lineetta vedrete tutte le linee da quella indicata in poi, se invece il numero è proposto alla linea vedrete tutte le istruzioni dall'inizio del programma fino a quella indicata.

Esempi

LIST	Mostra tutto il programma
LIST 100	Mostra la linea 100
LIST 100-	Mostra tutte le linee dalla 100 in poi
LIST -100	Mostra tutte le linee fino alla 100
LIST 5-20	Mostra tutte le linee dalla 5 alla 20

LOAD LOAD "nome del file" [,periferica] [,flag]
Esempio LOAD "CIAO", 8

Questo comando serve a trasferire in memoria un programma che si trova su cassetta o su disco. Se battete solo LOAD e poi schiacciate il RETURN lo schermo del Commodore 16 diventa vuoto, schiacciate "PLAY" e il Commodore 16 inizierà a cercare i programmi sulla cassetta. Quando ne ha trovato uno scriverà sullo schermo FOUND e poi il nome del file. Se è quello che volevate caricare schiacciate il tasto C= se no premete la barra spaziatrice fino a quando non si trova quello giusto. Una volta che il programma è caricato potete decidere se farlo girare <RUN > o averne un LISTato.
Potete anche far seguire al comando LOAD il nome di un determinato file fra virgolette ed eventualmente, dopo una virgola il numero della periferica dove questo è contenuto. In mancanza di numeri il computer cerca il programma nel registratore (che corrisponde al numero 1).

Esempi

LOAD	Cerca uno dopo l'altro tutti i programmi contenuti nella cassetta
LOAD "CIAO"	Cerca un programma il cui nome è contenuto fra virgolette
LOAD "CIAO", 8	Cerca un programma chiamato "CIAO" nel drive (8) LOAD può anche essere inserito in un programma per far caricare e leggere un altro programma detto "concatenato".

Il flag (o RELOCATE FLAG, flag di locazione) determina il punto della memoria dove il programma è caricato. Un flag 0 dice al Commodore 16 di caricare il programma all'inizio dell'area riservata ai programmi mentre un flag 1 dice di caricarlo dal punto in cui era stato salvato. In mancanza di specificazioni il valore del flag è ritenuto zero. Il flag è comunque utilizzato in genere solo quando si lavora con programmi redatti in linguaggio macchina.

NEW NEW

Questo comando cancella tutti i programmi in memoria e tutte le variabili usate. A meno che il programma non sia stato salvato su qualche supporto esterno una volta scritto NEW viene perduto, può anche essere usato come istruzione all'interno di un programma BASIC: quando il Commodore 16 arriva a questa istruzione il programma si cancella, normalmente non è di grande utilità ma può servire (per esempio per impedire l'accesso di intrusi).

RENAME RENAME "vecchio nome" TO "nuovo nome" [,Ddrive] [,Uunità]

Esempio RENAME "CIAO" TO "EHILA"' ,D1,U9

Serve a cambiare il nome di un file su un disco senza ovviamente modificarne il contenuto.

RENUMBER RENUMBER [nuova linea di partenza] [,incremento] [,vecchia linea di partenza]

Esempio RENUMBER 20,20,15

La nuova linea di partenza è il numero che la prima istruzione della parte del programma presa in esame avrà dopo la rinumerazione, in mancanza di indicazioni è 10.
Il secondo numero, l'incremento, è l'intervallo fra i numeri delle diverse istruzioni (10 20 30 40 etc) in mancanza di indicazioni è anch'esso 10. Il terzo rappresenta la vecchia linea di partenza, il numero da cui partire, in mancanza di indicazioni è la prima linea del programma. Le virgole devono essere inserite anche se i valori sono quelli di default (previsti dal computer).

Esempio

RENUMBER ,,,	Rinumera tutte le linee del programma partendo da 10 e con incrementi di 10
RENUMBER 20,20, 15	Rinumera le linee dalla 15 in poi dando alla prima il valore 20, alla seconda 40, alla terza 60 e così via.
RENUMBER,,65	Rinumera le linee dalla 65 in poi dando alla prima il valore 10 e con incrementi di 10

RUN RUN [numero linea]
Esempio RUN 100

Una volta che il programma è stato caricato nella memoria (con il comando LOAD) il comando RUN lo fa "girare" o lavorare. RUN cancella tutte le variabili del programma precedente ma non le istruzioni (in mancanza di un NEW quindi si rischia di lavorare con un programma composto da istruzioni provenienti da diversi programmi).

Se RUN non è seguito da nessun numero il computer parte dalla prima istruzione (e continua seguendo il loro numero d'ordine) altrimenti l'esecuzione parte da quel numero. RUN può essere usato da programma.

SAVE SAVE [''nome del file''] [,periferica] [,flag di fine cassetta]

Esempio SAVE ''CIAO'',1,1

Questo comando salva il programma in memoria su di un supporto magnetico. Se vi limitate a digitare SAVE e a premere RETURN il Commodore 16 tenterà di memorizzare il programma sul registratore a cassette. Purtroppo con le Cassette il Commodore 16 non ha nessun modo per controllare se in quella posizione esiste già un programma per cui state attenti.
Se dopo SAVE scrivete un nome fra virgolette o una stringa di variabili questo sarà il nome del file. Se volete potete specificare il numero della periferica facendo seguire il nome (e le virgolette) da una virgola e da un numero 1 indica il registratore a cassette e 8 il disk driver. Dopo questo ci può esser un altra virgola ed un secondo numero (0 o 1). Se questo è 1 il Commodore considera il programma salvato l'ultimo della cassetta. Questo numero inserisce infatti il flag di fine cassetta (End Of Tape flag). Se in seguito tenterete di caricare un programma e il Commodore 16 incontra uno di questi segnali avrete un FILE NOT FOUND ERROR

Esempi

SAVE	Salva il programma su cassetta senza nome
SAVE ''CIAO''	Salva il programma col nome CIAO
SAVE A$	Salva su cassetta il programma il cui nome è nella variabile A$
SAVE ''CIAO'', 1,1	Salva su cassetta il programma col nome CIAO e pone un flag di fine cassetta al termine del programma

SCRATCH SCRATCH ''nome del file'' [,Ddrive] [,Uunità]

Cancella un file dalla directory del disco. Come precauzione verrà richiesto ''Are You sure?''. Prima che il vostro Commodore 16 completi l'operazione. Scrivete ''Y'' per effettuare la cancellazione o premete 'N' per annullare l'operazione. Usate questo comando per cancellare files indesiderati, per creare più spazio su disco.

ESEMPIO:

SCRATCH"PIPPO",D1 Cancella il file "PIPPO" dal disco nel drive 1

VERIFY VERIFY "nomefile" [,numero dispositivo] [flag di rilocazione]

Questo comando fa in modo che il vostro Commodore 16 confronti il programma su cassetta o su disco con quello in memoria. Questo vi assicura che il programma appena salvato non contiene errori. Questo comando è anche molto utile per fare in modo che il computer posizioni il nastro alla fine dell'ultimo programma registrato. Tutto quello che dovete fare è di dire al vostro Commodore 16 di verificare il nome dell'ultimo programma sul nastro, cosa che il computer farà, dicendovi che il programma non contiene errori, cosa che già sapevate. Ora il nastro è nella posizione che volevate e potete registrare il prossimo programma senza paura di cancellarne uno vecchio.
VERIFY senza nient'altro fa in modo che il Commodore 16 controlli il prossimo programma senza badare al nome.
VERIFY seguito dal nome del programma tra virgolette o da una variabile stringa cerca il programma sul registratore e lo controlla.
VERIFY seguito da un nome, una virgola e un numero controlla il programma sul dispositivo con tale numero (1 per il registratore, 8 per il disco). Il flag di rilocazione è lo stesso che nel comando di LOAD.

ESEMPIO:

VERIFY	Controlla il prossimo programma sul registratore.
VERIFY"REALTÀ"	Cerca il programma REALTÀ sul registratore e lo confronta con quello in memoria
VERIFY "ME",8,1	Cerca il file "ME" sul disco e lo controlla

ISTRUZIONI BASIC

BOX BOX [numero colore,] a1,b1,[a2,B2] [,angolo] [,riempimento]

Numero colore	Numero colore (0..3); all'accensione è 1 (colore di sfondo)
a1,b1	Coordinate di un vertice (in scala)
a2,b2	Coord. del vertice opposto ad a1,b1 (in scala); in mancanza di istruzioni è il Pixel Cursor.
Angolo	gradi di rotazione in senso orario; normalmente è 0.
Riempimento	Se uno riempie la figura col colore scelto, se zero no. Normalmente è 0.

Descrizione del comando BOX.

Questo comando vi permette di disegnare un rettangolo di qualsiasi misura dovunque sullo schermo. Per ottenere il valore normale, mettere una virgola senza aggiungere il valore. La rotazione avviene rispetto al centro del rettangolo. Il Pixel Cursor (PC) è lasciato ad a2,b2 dopo che l'istruzione BOX è stata eseguita.

ESEMPI:

BOX 1 , 10 , 10 , 60 , 60	Disegna il contorno di un rettangolo
BOX , 10 , 10 , 60 , 60 , 45 , 1	Disegna un rettangolo riempito e ruotato
BOX , 30 , 90 , , 45 , 1	Disegna un poligono riempito e ruotato

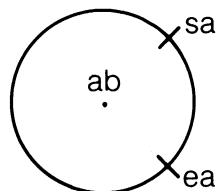
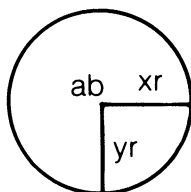
CHAR CHAR [numero colore],x,y ,"stringa" [,flag di reverse]

Numero colore	Numero del colore (0..3)
x	Colonna del carattere (0..39)
y	Riga del carattere (0..24)
Stringa	Stringa da stampare
Flag di reverse	0=Stringa in positivo, 1=Stringa in negativo

Un testo (una stringa alfanumerica) può essere visualizzato sullo schermo alla posizione specificata dal comando CHAR. I dati del carattere sono letti dall'area della ROM dei caratteri plus/4. Dovete fornire le coordinate x e y della posizione di partenza della stringa che volete visualizzare; il colore ed il flag di inversione sono opzionali.
La stampa della stringa viene continuata sulla riga successiva se oltrepassa il margine destro dello schermo. Quando usate il modo TEXT, la stringa stampata dal comando CHAR si comporta esattamente come una istruzione PRINT, inclusi i campi inversi, controlli di cursore e di lampeggio, etc. Questi controlli all'interno della stringa non funzionano quando il comando CHAR è usato all'interno del modo grafico.

Nota: Quando siete nel modo MULTICOLORE per visualizzare un carattere in MULTICOLORE2 porre il numero del colore a 0 e il flag di inversione a 1. Per visualizzare un carattere in MULTICOLORE1 porre il numero del colore a 0 e il flag di inversione a 0.

CIRCLE CIRCLE [cs] , [a,b] , xr [, [yr] [, [sa] [, [ea] [, [angolo] [,inc]]]]

cs	numero colore (0 ... 3)
a,b	coordinate del centro (in scala); normalmente corrisponde al PC
xr	raggio x (in scala)
yr	raggio y (normalmente uguale a xr)
sa	angolo di partenza (normalmente 0)
ea	angolo di arrivo (normalmente 360)
angolo	angolo di rotazione in senso orario espresso in gradi normalmente=0
inc	numero di gradi tra i segmenti (normalmente 2)

Descrizione del comando CIRCLE

Con il comando CIRCLE potete disegnare un cerchio, un'ellisse, un arco, un triangolo o un ottagono. Le coordinate finali sono sulla circonferenza alla fine dell'arco disegnato specificato dal parametro "ea". Ogni rotazione è riferita al centro. Definendo il raggio x uguale al raggio y, non verrà disegnato un cerchio poiché le coordinate x e y hanno scala diversa. Gli archi sono disegnati dall'angolo di partenza a quello di arrivo specificati rispettivamente dai parametri "sa" e "ea". Il parametro "inc" definisce la definizione della figura; valori bassi disegneranno una figura arrotondata

ESEMPI:

CIRCLE , 100 , 100 , 65 , 10	Disegnaun'ellisse
CIRCLE , 160 , 100 , 65 , 50	Disegna un cerchio
CIRCLE , 60 , 40 , 20 , 18 , , , , 45	Disegna un ottagono
CIRCLE , 260 , 40 , 20 , , , , , 90	Disegna un diamante
CIRCLE , 60 , 140 , 20 , 18 , , , , 120	Disegna un triangolo

CLOSE CLOSE numero file

Questo comando completa e chiude ogni file aperto dal comando OPEN. Il numero seguente la parola CLOSE è il numero del file da chiudere.

ESEMPIO:

CLOSE 1   Il file logico 1 è chiuso.

CLR CLR

Questo comando cancella tutte le variabili in memoria, ma lancia intatto il programma. Questo comando è eseguito automaticamente quando viene dato un comando di RUN o NEW, o quando viene effettuata una correzione del programma.

CMD CMD numero file

Il comando CMD invia i dati che normalmente andrebbero allo schermo (istruzioni di PRINT, LIST, ma non i POKE nella memoria di schermo) ad un'altra periferica. Questa potrebbe essere una stampante o un file di dati su nastro o disco. Questa periferica deve essere preventivamente aperta. Il comando CMD deve essere seguito da un numero o una variabile numerica che si riferisca al file.

ESEMPI:

OPEN 1,4	Apre la periferica numero 4, che è la stampante.
CMD 1	Tutti i dati in uscita vengono ora inviati alla stampante.
LIST	Il listato va sulla stampante, non sullo schermo (anche la parola READY).
PRINT 1	riporta l'uscita sullo schermo
CLOSE 1	Chiude il file.

COLOR COLOR Destinazione, numero colore [,luminanza]

Assegna un colore ad una delle cinque destinazioni.

Numero	Destinazione
O	Sfondo
1	Primo piano
2	Multicolore 1
3	Multicolore 2
4	Bordo

I colori possono essere scelti tra 1 e 16 (1=nero, 2=bianco, 9=arancio, etc., come i colori sulla vostra tastiera). Come opzione potete aggiungere il livello di luminanza, compreso tra 0 e 7, dove 0 è luminanza minima e 7 luminanza massima. Il valore normale è sette. La luminanza vi permette di selezionare otto livelli di brillantezza per ogni colore eccetto il nero.

DATA DATA lista di costanti separate da virgole.

Questa istruzione è seguita da una lista di valori da usarsi con l'istruzione READ. I valori possono essere numeri o parole e sono separati da virgole. Non è necessario includere le parole tra virgolette, a meno che contengano uno dei seguenti caratteri: spazio, due punti, virgola. Se tra due virgole non c'è alcun valore, esso sarà letto come 0 per un numero, o come stringa vuota. Vedere anche il comando RESTORE che permette al vostro Commodore 16 di rileggere i dati.

ESEMPIO:

DATA 100,200,FRED,"WILMA",3,LA,REG120

DEF FN (DEFinisce una FuNzione) DEF FN nome (variabile)=Espressione.

Una lunga formula è spesso usata in un programma questa istruzione vi permette di risparmiare spazio.
Il nome che date alla funzione comincia con le lettere FN, seguite da ogni nome legale di variabile numerica. Prima bisogna definire la funzione usando l'istruzione DEF seguita dal nome che avete dato alla funzione facendolo seguire da una serie di parentesi () con una variabile numerica (in questo caso X) inclusa.

Di seguito si ha un segno '=' seguito dalla formula che volete definire. La formula può essere richiamata sostituendo alla X un qualsiasi numero, usando il formato mostrato alla linea 20 nell'esempio seguente.

ESEMPI:

10 DEF FNA(X)=12*(34.75-X/.3)+X

20 PRINT FNA(7)

Il numero 7 è sostituito in ogni punto della formula dove compare la X nell'istruzione DEF

DIM DIM variabile (indice) [, variabile(indice)]...

Prima di poter usare delle matrici il programma deve eseguire un'istruzione di DIM per stabilire le DIMensioni di quella matrice. L'istruzione DIM può essere seguita dal nome della matrice, che può essere ogni possibile nome di variabile. Poi tra parentesi ponete il numero o la variabile numerica degli elementi, per ogni dimensione. Una matrice con più di un indice è una matrice multidimensionale. Si può usare un qualsiasi numero come indice della matrice, ma bisogna tener presente che l'intera lista di variabili che così create prende spazio in memoria, ed è facile uscire dalla memoria se non state attenti. Per comprendere il numero di variabili create con ogni DIM, moltiplicate tra di loro gli indici di ogni dimensione di una matrice. (Ogni matrice parte dall'elemento 0).

Nota: Una matrice intera occupa 2/5 dello spazio occupato da una matrice in virgola mobile delle stesse dimensioni.

ESEMPIO:

10 DIM A$(40),B7(15),CC%(4,4,4)

41 elementi, 16 elementi, 125 elementi.

Potete dimensionare più di una matrice con la stessa istruzione DIM separandole con la virgola. Se il programma esegue un'istruzione DIM più di una volta per la stessa matrice, darà il segnale di errore. È buona prassi di programmazione posizionare l'istruzione di DIM vicino all'inizio del programma.

DO/LOOP/WHILE/UNTIL/EXIT

DO [UNTIL argomento booleano / WHILE argomento booleano] istruzioni [EXIT]
LOOP [UNTIL argomento booleano / WHILE argomento booleano]
(Un esempio di argomento booleano è A=1 o H >=57).
Esegue le istruzioni comprese tra l'istruzione DO e l'istruzione LOOP. Se né UNTIL né WHILE modificano le istruzioni DO o LOOP, l'esecuzione delle istruzioni comprese continua indefinitamente. Se un'istruzione EXIT è compresa all'interno del ciclo DO, l'esecuzione è trasferita alla prima istruzione seguente l'istruzione LOOP. I cicli DO possono essere eseguiti seguendo le regole definite per i cicli FOR-NEXT. Se è usato il parametro UNTIL, il programma continua il ciclo finché l'argomento booleano è soddisfatto (diventa vero). Il parametro WHILE è fondamentalmente l'opposto del parametro UNTIL: il programma continua il ciclo finché l'argomento booleano è vero.

ESEMPIO:

```
DO UNTIL X=0 OR X=1
(istruzioni)
LOOP
DO WHILE A$"":GETA$:LOOP
```

DRAW

DRAW numero colore, a1,b1, TO a2,b2...

Con questo comando potete disegnare punti singoli, linee o figure. Dovete specificare il numero del colore (0..3), il punto di inizio (a1,b1) e i punti di fine (a2,b2).

ESEMPI:

Un punto:	DRAW 1,100,50	Nessun punto di fine specificato. Normalmente a2 e b2 assumono il valore di a1 e b1 per disegnare un punto singolo.
Linee:	DRAW ,10,10, TO 100,60	
	DRAW , TO 25,30	
Una figura:	DRAW ,10,10 TO 10,60 TO 100,60 TO 10,10	

END

Quando il programma esegue un'istruzione di END si ferma immediatamente. Potete usare il comando CONT per far ripartire il programma dall'istruzione seguente a quella di END.

FOR...TO...STEP FOR variabile = valore d'inizio TO valore di fine [STEP incremento]

Questa istruzione funziona con l'istruzione NEXT per ceare una sezione di programma che si ripete per un certo numero di volte. Potete fare in modo che il computer ripeta un gran numero di volte un ciclo chiuso per creare una pausa di qualche secondo, nel caso abbiate bisogno di contare qualcosa, o dobbiate fare qualcosa per un certo numero di volte.
La variabile del ciclo è quella che viene incrementata o sottratta per la durata del ciclo FOR-NEXT. Il valore iniziale e quello finale segnano l'inizio e la fine del conteggio per la variabile del ciclo.
La logica dell'istruzione FOR è la seguente: per prima cosa viene assegnato il valore iniziale alla variabile del ciclo. Quando il programma raggiunge una linea con l'istruzione NEXT, il valore dell'incremento di STEP (normalmente uguale a 1) viene sommato alla variabile del ciclo, e si controlla se il suo valore supera quello finale. Se non è più alto, la linea successiva eseguita è quella immediatamente dopo il FOR. Se la variabile del ciclo è più grande del valore finale, l'istruzione successiva eseguita è quella seguente il NEXT. Vedere anche l'istruzione NEXT.

ESEMPI:

```
10 FOR L=1 TO 20
20 PRINT L
30 NEXT L
40 PRINT"CIAO L="L
```

Questo programma stampa i numeri da 1 a 20 sullo schermo seguiti dal messaggio CIAO L=21.
Il valore finale del ciclo può essere seguito dalla parola STEP ed un altro numero o variabile. In questo caso il valore seguente lo STEP è sommato ogni volta che viene incontrato il NEXT alla variabile del ciclo. Questo permette di contare all'indietro, per frazioni o come è necessario.
È possibile racchiudere un ciclo FOR-NEXT dentro l'altro. Questa procedura è conosciuta come "annidamento". Bisogna fare attenzione nell'"annidare" i cicli, poiché l'ultimo ciclo aperto deve essere il primo che viene chiuso.

Esempio di ciclo nidificato:

```
10 FOR L=1 TO 100
20 FOR A=5 TO 11 STEP 2
30 NEXT A
40 NEXT L
```

Questo ciclo FOR-NEXT è annidato in quello più grande.

GET GET lista di variabili

L'istruzione GET è un modo per acquisire dalla tastiera un carattere alla volta. Quando viene eseguita una GET, il carattere corrispondente al tasto premuto è acquisito. Se non era stato schiacciato alcun tasto, viene assegnato un carattere vuoto, e il programma continua senza aspettare la pressione di un altro tasto. Non c'è bisogno di premere il tasto RETURN ; infatti il tasto RETURN può essere acquisito con la GET.
La parola GET è seguita da un nome di variabile, normalmente una variabile stringa. Se viene usata una variabile numerica e viene premuto un tasto non numerico, il programma si fermerà con un messaggio di errore. L'istruzione GET può anche essere inserita in un ciclo che controlla se il valore acquisito è nullo, in modo da aspettare che un tasto venga premuto per continuare il programma. L'istruzione GETKEY può anche essere usata in questo caso. Questo comando può essere eseguito solo in un programma.

ESEMPIO:

10 GET A$:IF A$<>"A" THEN 10 Questa linea aspetta che venga premuto il tasto A

GETKEY GETKEY lista di variabili

L'istruzione GETKEY è molto simile alla GET. A differenza dell'istruzione GET, GETKEY attende che venga premuto un tasto. Questo permette di usarla per aspettare un singolo carattere dalla tastiera.
Questo comando può essere eseguito solo in un programma.

ESEMPIO:

10 GETKEY A$

Questa linea aspetta che sia premuto un tasto. Battendo un tasto qualsiasi il programma riprenderà.

GET # GET # numero del file, lista di variabili

Usata con una periferica preventivamente aperta o con un file, permette di acquisire un carattere alla volta. In altre parole funziona come un'istruzione GET. Questo comando può essere usato solo in un programma.

ESEMPIO:

GET # 1,A$

GOSUB GOSUB Numero di linea

Questa istruzione funziona come la GOTO, con l'unica differenza che il vostro Commodore 16 ricorda da dove viene. Quando il programma incontra una linea con l'istruzione RETURN salta indietro all'istruzione immediatamente successiva alla GOSUB. La parte di programma puntata dalla GOSUB viene chiamata "subroutine" o "sottoprogramma". Una subroutine è comoda quando esiste una parte di programma che deve essere usata più volte da diverse porzioni del programma. Invece di riscrivere la parte di programma ogni volta, la potete porre in una subroutine e saltare ad essa con l'istruzione GOSUB dalle diverse parti del programma. Vedi anche l'istruzione RETURN.

ESEMPIO:

20 GOSUB 800	Significa andare alla subroutine che comincia alla linea 800 ed eseguirla.

800 PRINT "CIAO!":RETURN

GOTO o GO TO Dopo aver eseguito l'istruzione GOTO la prossima linea sarà quella col numero di linea indicato. Quando usata nel modo diretto, l'istruzione GOTO vi permette di iniziare l'esecuzione del programma alla linea specificata senza cancellare le variabili.

ESEMPIO:

```
10 PRINT "CIAO"
20 GOTO 10
```

L'istruzione GOTO nella linea 20 fa in modo che la linea 10 sia eseguita ininterrottamente fino alla pressione del tasto RUN/STOP.

GRAPHIC GRAPHIC modo[,opzione di cancellazione]

Questa istruzione pone il vostro Commodore 16 in uno dei 5 modi grafici.

Modo	Descrizione
0	Testo normale
1	Alta risoluzione grafica
2	Alta risoluzione grafica mista a testo
3	Grafica multicolore
4	Grafica multicolore mista a testo

Quando viene eseguita l'istruzione GRAPHIC 1..4, vengono allocati 10 kbyte di memoria per lo schermo grafico, e l'inizio della memoria BASIC per i programmi viene spostata dopo tale area di alta risoluzione. Quest'area rimane allocata anche se si ritorna in modo testo (GRAPHIC 0). Se viene posto 1 come secondo argomento lo schermo viene anche cancellato. Eseguendo l'istruzione GRAPHIC CLR viene resa nuovamente disponibile al BASIC l'area di 10 kbyte usata dai modi grafici.

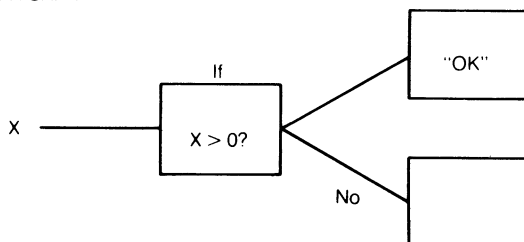
ESEMPI:

GRAPHIC 3,1	Seleziona il modo grafico multicolore e cancella lo schermo.
GRAPHIC CLR	Cancella l'area grafica e la restituisce al BASIC

IF... THEN [..:ELSE]

IF espressione THEN istruzione [:ELSE istruzione]

L'istruzione IF...THEN fa analizzare al computer un'espressione BASIC preceduta da IF e che comporti uno solo dei due possibili risultati seguenti. Se il risultato è vero, l'istruzione seguente la parola THEN viene eseguita. Questa espressione può essere una qualsiasi istruzione BASIC. Se l'espressione è falsa, il programma passa direttamente alla linea successiva, a meno che sia presente un'istruzione ELSE. L'espressione da valutare può essere una variabile o una formula, nel qual caso è considerata vera se diversa da 0 e falsa se uguale a 0. Nella maggior parte dei casi c'è un'espressione comprendente gli operatori di relazione =,<, >, <=, >=,<>,AND,OR,NOT).
L'istruzione ELSE, se presente deve essere sulla stessa linea della IF...THEN. Quando un'istruzione ELSE è presente, viene eseguita quando non è eseguita l'istruzione THEN. In altre parole, l'istruzione ELSE è eseguita quando l'espressione IF è falsa.

FIGURA FLOWCHART

ESEMPIO:

50 IF X>0 THEN PRINT"OKAY":ELSE END

Controlla il valore di X. Se X è maggiere di 0 viene eseguita l'istruzione dopo il THEN, altrimenti viene eseguita quella dopo l'ELSE.

INPUT INPUT "stringa da stampare"; lista variabili

L'istruzione INPUT permette al computer di chiedere dati all'utente e di porli in una o più variabili. Il programma si ferma, stampa "?" sullo schermo e aspetta che l'utente batta la risposta e batta il tasto RETURN. La parola INPUT è seguita da un nome di variabile o da una lista di nomi separati da virgole. Ci può essere un messaggio tra virgolette prima della lista di variabili da introdurre. Se questo messaggio è presente deve esserci un ";" dopo la chiusura delle virgolette. Quando più di una variabile deve essere inserita, dovranno essere separate da virgole al momento della battitura, altrimenti il computer chiederà i valori rimanenti stampando "??". Se schiacciate il tasto RETURN senza aver introdotto alcun valore, la variabile di INPUT manterrà il valore che aveva prima. Questa istruzione può essere eseguita solo in programma.

ESEMPIO:

10 INPUT "QUAL'È IL TUO NOME";A$
20 INPUT "QUANTO FA 2+2";A

INPUT # INPUT# numero del file, lista di variabili.

Questa istruzione funziona come la INPUT ma prende i dati da una periferica o da un file preventivamente aperti. Non è possibile stampare messaggi. Questo comando può essere usato solo in programma.

ESEMPIO:

INPUT # 2,A$,C$,G

LET [LET] variabile = espressione

La parola LET è difficilmente usata in un programma perché come tale non è necessaria, ma l'istruzione stessa è il cuore di tutti i programmi BASIC. Ogni volta che una variabile è definita, l'istruzione LET è sempre sottintesa. Il nome della variabile che deve ricevere il risultato di un calcolo è sulla sinistra del segno "=" e i numeri o la formula sono sulla destra.

ESEMPIO:

```
10 LET A=5
20 B=6
30 C=A*B+3
40 B$="CIAO!"
```

Nelle linee 20,30 e 40 LET è sottinteso (non è necessario scriverlo)

LOCATE LOCATE coordinata x, coordinata y

Il comando locate pone il Pixel Cursor (PC) dovunque sullo schermo. Il PC è la locazione corrente di inizio esecuzione della prossima istruzione di disegno. A differenza del normale cursore, il PC non è visibile, ma potete muoverlo col comando LOCATE.

ESEMPIO:

LOCATE 160,100

Posiziona il PC nel centro dello schermo ad alta risoluzione. Non si vedrà nulla finché non verrà disegnato qualcosa. Si può sapere dove si trova il PC tramite la funzione RDOT (0) che restituisce la coordinata X e RDOT(1) che restituisce la coordinata Y. Il colore del punto su cui si trova il PC può essere rilevato con la funzione RDOT(2). (In tutti i comandi di disegno dove è disponibile un'opzione colore, si può scegliere un valore da 0 a 3, dove 0 è lo sfondo, 1 il primo piano, 2 il multicolore 1, 3 il multicolore 2 come sorgente di colore).

MONITOR MONITOR

Questo comando vi porta fuori dal BASIC all'interno di un monitor in linguaggio macchina. Il MONITOR è usato per sviluppare, correggere ed eseguire programmi in linguaggio macchina più facilmente che da BASIC. Vedere la sezione sui comandi del monitor per maggiori informazioni. (Se volete uscire dal MONITOR battete X e premete RETURN e tornate in BASIC).

NEXT NEXT[variabile,...,variabile]

L'istruzione NEXT è usata con l'istruzione FOR. Quando il computer incontra un'istruzione NEXT torna alla corrispondente istruzione FOR e controlla la variabile di ciclo. (Vedere l'istruzione FOR per maggiori dettagli). Se il ciclo è finito, l'esecuzione procede con l'istruzione successiva alla NEXT. La parola NEXT può essere seguita da una o più variabili separate da virgole o da nessuna variabile. Se non ci sono variabili, viene chiuso l'ultimo ciclo aperto. Se le variabili sono specificate, vengono considerate da sinistra a destra.

ESEMPIO:

```
10 FOR L=1 TO 10:NEXT
20 FOR L=1 TO 10:NEXT L
30 FOR L=1 TO 10:FOR M=1 TO 10:NEXT M,L
```

ON ON espressione <GOTO/GOSUB> numero linea [,numero linea,...]

Questo comando può porre le istruzioni GOTO e GOSUB in una speciale versione dell'istruzione IF. La parola ON è seguita da una formula e da GOTO o GOSUB, e una lista di numeri di linea separati da virgole. Se il risultato dell'espressione è uno, viene eseguita la prima linea della lista. Se il risultato è due viene eseguita la seconda e così via.
Se il risultato è zero, o più grande del numero delle linee disponibili nella lista, la prossima linea eseguita è l'istruzione seguente ON. Se il numero è negativo, avverrà una segnalazione di ILLEGAL QUANTITY ERROR.

ESEMPIO:

```
10 INPUT X:IF X<0 THEN 10
20 ON X GOTO 50,30,30,70
25 PRINT "GIUSTO!":GOTO 10
30 PRINT "TROPPO GRANDE!":GOTO 10
50 PRINT "TROPPO BASSO":GOTO 10
70 END
```

Quando X=1, ON passa il controllo al primo numero di linea della lista.
Quando X=2 ON passa il controllo alla seconda linea della lista (30). etc.

OPEN OPEN numero file, numero periferica [,indirizzo secondario] [, "nomefile, tipo, modo"]

L'istruzione OPEN permette al vostro Commodore 16 di accedere a periferiche come il registratore a cassette e il disk driver, alla stampante, o anche allo schermo video. La parola OPEN è seguita da un numero di file logico, che è il numero al quale tutte le altre istruzioni BASIC si riferiranno. Questo numero va da 1 a 255. C'è sempre un secondo numero dopo il primo, chiamato numero della periferica. Il numero di periferica 0 corrisponde alla tastiera del Commodore 16, il numero 3 allo schermo, l'1 al registratore, il 4 alla stampante e l'8, normalmente, al driver. Uno 0 può essere posto davanti al numero della periferica (per esempio 08 per 8, che per il Commodore 16 hanno la stessa funzione). Usare come numero di file lo stesso numero che contrassegna di periferica è una buona idea. Dopo il secondo numero ci può essere un terzo numero chiamato indirizzo secondario. Nel caso del registratore, questo può essere 0 per la lettura, 1 per la scrittura e 2 per scrivere il marcatore di fine nastro. Nel caso del disco, il numero si riferisce al numero di canale. Nella stampante gli indirizzi secondari sono usati per definire i modi di funzionamento della stampante. Il COMMODORE 16 PROGRAMMER'S REFERENCE MANUAL o il manuale di ogni specifica periferica contengono maggiori informazioni sugli indirizzi secondari. Ci può essere anche una stringa dopo il terzo numero che può essere un comando per il disco, o il nome del file sul registratore o sul disco. Il tipi e il modo si riferiscono solamente a files su disco. (I tipi di files sono: PRG, SEQ, REL e USR; i modi sono READ e WRITE).

ESEMPI:

10 OPEN 3,3	Definisce lo schermo come periferica.
20 OPEN 1,0	Definisce la tastiera come periferica.
20 OPEN 1,1,0,"CIAO!"	Definisce il registratore come periferica in lettura, il file da cercare è CIAO!
OPEN 4,4	Apre un canale per usare la stampante.
OPEN 15,8,15	Apre il canale di comando del disco
5 OPEN 8,8,12,"CIAO2,SEQ,WRITE"	Crea un file sequenziale in scrittura su disco

Vedere anche: CLOSE, CMD, GET #, INPUT#, PRINT # e le variabili di sistema ST, DS e DS$.

PAINT PAINT [numero colore] [,[a,b] [,modo]

Numero colore	(O..3); Normalmente 1 (colore del primo piano)
a,b	Coordinate di partenza in scala (normalmente è il PC)
Modo	0= riempie una figura definita dal numero colore selezionato 1= riempie un'area definita da qualsiasi colore non di sfondo

Il comando PAINT vi permette di riempire un'area con un colore. Riempie l'area attorno al punto specificato compresa in un bordo dello stesso colore (o un qualsiasi colore non di sfondo, a seconda del modo che avete scelto).
La posizione finale del PC rimarrà invariata (a,b).

Nota: Se il punto di partenza è già del colore prescelto (o un qualsiasi colore non di sfondo quando si è in modo 1), non ci saranno cambiamenti.

ESEMPIO:

10 CIRCLE 160,100,65,50	Disegna il contorno di un cerchio.
20 PAINT 160,100	Riempie il cerchio di colore.

POKE POKE indirizzo, valore

Il comando POKE vi permette di cambiare qualsiasi valore nella memoria RAM del vostro Commodore 16, e vi permette di modificare la maggior parte dei registri di ingresso/uscita del Commodore 16. La parola POKE è sempre seguita da due numeri o espressioni. Il primo numero è una locazione della memoria del computer. Questo può avere un qualsiasi valore che può andare da 0 a 65535. Il secondo numero è un valore da 0 a 255, il quale è posto nella locazione, sostituendo ogni valore precedente. Questa istruzione può anche controllare il contenuto dello schermo, premettendovi di inserire un qualsiasi carattere in qualsiasi punto della memoria video.

ESEMPIO:

10 POKE 28000,8 Mette nella locazione 28000 il valore 8

NOTA: Anche PEEK, un comando legato a POKE, è descritto tra le FUNZIONI.

PRINT PRINT lista di stampa

L'istruzione PRINT è la principale istruzione BASIC di uscita. Per la maggior parte delle persone è la prima che si impara, ma pochi la sanno sfruttare appieno. La parola PRINT può essere seguita da:

Caratteri tra virgolette	("Linee di testo")
Nomi di variabili	(A,B,A$,X$)
Funzioni	(SIN(23),ABS(-33))
Segni di punteggiatura	(; ,)

I caratteri fra virgolette sono spesso chiamati "letterali". Poiché vengono stampati esattamente come appaiono. Dei nomi di variabili vengono stampati i valori contenuti (anche un numero o una stringa). Delle funzioni viene stampato il loro valore numerico. I segni di punteggiatura sono usati per stabilire il formato dei dati in uscita. La virgola divide lo schermo in quattro colonne per la stampa dei dati mentre ";" non aggiunge alcuno spazio. Anche "." può essere usato come gli altri simboli nell'istruzione PRINT. Fa si che la prossima PRINT sia l'esatta continuazione della precedente.

ESEMPIO:

10 PRINT"CIAO"	CIAO
20A$="SALVE":PRINT"CIAO,"A$	CIAO,SALVE
30 A=4:B=2:PRINT A+B	6
50 J=41:PRINT J;:PRINT J-1	41 40
60 C=A+B:D=A-B:PRINT A;B;C,D	4 2 6 2

Vedere anche le funzioni POS(), SPC() e TAB().

PRINT # PRINT # numero file, lista di stampa

Ci sono alcune differenze tra questa istruzione e la PRINT. Prima di tutto la parola PRINT # è seguita da un numero, il quale si riferisce alla periferica o al file di dati preventivamente aperto. Il numero è seguito da una virgola e da una lista di cose da stampare. "," e ";" funzionano nello stesso modo dell'istruzione PRINT. Alcune periferiche non possono lavorare con TAB e SPC.

ESEMPIO:

```
PRINT # 1,"CIAO!!!",A$,B$,
```

PRINT USING PRINT [numero file] USING lista del formato; lista di stampa;

Questa istruzione vi permette di definire il formato di stampa di stringhe e valori numerici, sia su schermo che su stampante o su altre periferiche. Bisogna porre il formato desiderato tra virgolette, (e questa è la "lista" del formato); poi bisogna aggiungere un ";" e la lista di ciò che si vuole stampare.

ESEMPIO:

```
5 X=32:Y=100.23:A$="GU!"
10 PRINT USING "$# # . # #";13.25,X,Y
20 PRINT USING "# # #  > #";"CBM",A$
```

Quando date il RUN, la linea 10 stampa:

$13.25$32.00$*****

> Stampa ***** al posto del valore di Y poiché era formato da 5 cifre, non conforme, quindi, alla lista di formato (come spiegato sotto)

La linea 20 stampa questo:

CBM GU! — Lascia tre spazi prima di stampare CBM, come definito nella lista di formato.

Carattere	**Numerico**	**Stringa**
(#)	X	X
Più (+)	X	
Meno (-)	X	
Punto decimale (.)	X	
Virgola (,)	X	
Segno del dollaro ($)	X	
Quattro frecce (↑↑↑↑)	X	
Segno uguale (=)		X
Segno maggiore (>)		X

"#" lascia lo spazio per un carattere singolo nel campo di uscita. Se i dati contengono più caratteri del numero di "#" nel vostro campo di formato succede che: per un valore numerico, l'intero campo è riempito con asterischi (*). Nessun numero è stampato.

ESEMPIO:

10 PRINT USING "# # # #",X

Per questi valori di X il formato sarà il seguente:

X=12.34	12
X=567.89	568
X=123456	****

Per una stringa, il dato è troncato al valore del campo. Verranno cioè stampati tanti caratteri quanti sono i segni "#", e il troncamento avviene a destra.
I segni "+" e "-" possono essere usati o all'inizio o alla fine di un campo di formato. Il segno "+" è stampato se il numero è positivo, il segno "-" è stampato se il numero è negativo.
Se usate un segno "-" e il numero è positivo, al suo posto verrà stampato un carattere di blank.
Se non usate nessuno dei due segni nel vostro campo di formato per un dato numerico, un segno "-" è stampato prima della prima cifra o del simbolo "$" se il numero è negativo, altrimenti nulla se il numero è negativo. Questo significa che potete stampare un carattere in più se il numero è positivo. Se ci sono troppe cifre da porre nel campo specificato da "#", "+" e "-", allora avviene un overflow e il campo viene riempito da "*"

Un punto decimale (.) designa la posizione del punto decimale del numero. Potete avere solo un punto decimale per ogni campo di formato. Se non specificate un punto decimale nel vostro campo di formato, il valore è arrotondato all'intero più vicino, e stampato senza cifre decimali.
Quando specificate il punto decimale, il numero di cifre precedenti il punto stesso (incluso il segno "-" se il numero è negativo) non devono superare il numero di "//" prima del punto decimale. Se ci sono troppe cifre avviene un overflow, ed il campo viene riempito di "*".
Una virgola (,), vi permette di piazzare una virgola nel campo numerico. La posizione della virgola nella lista di formato indica dove appare la virgola nel numero da stampare. Solo le virgole all'interno di un numero vengono stampate. Virgole non usate sulla sinistra della prima cifra appaiono come carattere riempitore. Almeno un "#" deve precedere la prima virgola.
Se specificate le virgole in un campo, ed il numero è negativo, allora un segno "-" è stampato come primo carattere, anche se il carattere della posizione è specificato come una virgola.

ESEMPI:

Campo	**Espressione**	**Risultato**	**Commento**
##.#+	-.01	0.01-	Viene aggiunto lo zero prima del punto.
## .#-	1	1.0	Viene aggiunto lo zero dopo il punto.
####	-100.5	-101	Il numero è arrotondato.
####	-1000	****	Il numero più il segno sono in eccesso.
###.	10	10.	Viene aggiunto il punto decimale.
#$##	1	$1	Viene aggiunto il simbolo "$".

Un simbolo "$" indica che tale simbolo verrà stampato nel numero. Se volete che il segno "$" sia sempre piazzato prima del numero, ci vuole sempre un segno "#" prima di "$". Se specificate il segno "$" senza aggiungere "#" davanti, il segno "$" è stampato nella posizione mostrata nel campo.
Se specificate "," e/o "+" oppure "-" nel campo di formato con "$", il vostro programma stamperà una virgola o il segno prima di "$".
I quattro segni riempitori (freccia in sù) sono usati per specificare che il numero deve essere stampato con l'esponenziale (E+il formato). Dovete usare "#" assieme ai caratteri per specificare l'ampiezza del campo. I caratteri possono apparire anche prima o dopo il "#" nel campo di formato.
Dovete specificare i quattro caratteri (↑↑↑↑) quando volete stampare un numero in notazione scientifica (E+il formato). Se specificate più di uno ma meno di quattro frecce otterrete un SINTAX ERROR. Se ne specificate più di quattro solo i primi quattro saranno usati. Il primo carattere riempitore è intepretato come un simbolo non di testo.

Un segno "=" è usato per centrare una stringa in un campo. Voi specificate l'ampiezza del campo attraverso il numero di caratteri (#e=) nel campo di formato. Se la stringa contiene più caratteri di quanti possano essere inseriti nel campo, i caratteri più a destra sono troncati e la stringa riempie l'intero campo. Il segno ">" è usato per giustificare a destra una stringa in un campo. Se la stringa contiene meno caratteri dell'ampiezza del campo, la stringa viene giustificata nel campo. Gli eventuali eccedenti l'ampiezza del campo sono troncati da destra.

PUDEF PUDEF "Da 1 a 4 caratteri"

PUDEF vi permette di ridefinire fino a 4 simboli nell'istruzione PRINT USING. Potete cambiare gli spazi vuoti, le virgole al posto dei punti decimali e "$" in altri caratteri ponendo il nuovo carattere nella posizione corretta nella stringa di controllo dell'istruzione PUDEF.
La posizione 1 è il carattere riempitore. Normalmente è uno spazio vuoto. Piazzate un nuovo carattere qui quando volete che appaia un altro carattere al posto del carattere vuoto.
La posizione 2 è il carattere ",". Normalmente è ",".
La posizione 3 è ".".
La posizione 4 è "$".

ESEMPI:

10 PUDEF "*"	Stampa ""*" al posto dello spazio vuoto.
20 PUDEF "&"	Stampa "&" al posto di ",".
30 PUDEF ".,"	Stampa ""." al posto di "," e "," al posto di ".".

READ READ lista di variabili.

Questa istruzione è usata per prendere informazioni dall'istruzione DATA e porli nelle variabili per essere usati. La lista di variabili dell'istruzione DATA può contenere sia stringhe che numeri. Bisogna far attenzione a non leggere una stringa quando l'istruzione READ aspetta un numero, cosa che produce un messaggio di errore.

ESEMPIO:

READ A$,G$,A5

REM
REM Messaggio

L'istruzione REM permette di inserire note all'interno del programma. Cio può servire a spiegare la sezione di un programma, dare informazioni sull'autore, etc. L'istruzione REM non ha alcun effetto durante il programma, eccetto allungarlo e dunque rallentarlo. La parola REM può essere seguita da qualsiasi testo, sebbene l'uso di caratteri grafici dia strani risultati.

ESEMPIO:

10 NEXT:REM COMMENTO

RESTORE RESTORE [numero linea]

Quando viene eseguita in un programma, il puntatore posizionato sul prossimo valore di un'istruzione DATA che deve essere letto, riportano al primo valore della lista. Questo vi dà modo di rileggere (READ) le informazioni. Se un [numero linea] segue l'istruzione RESTORE, il puntatore viene posizionato a quella linea. Altrimenti il puntatore è rioportato alla prima istruzione DATA del programma.

ESEMPIO:

RESTORE 200

RESUME RESUME [Numero linea/NEXT]

Usato per riportare l'esecuzione dopo che si è verificato un errore. Senza argomenti, RESUME riesegue la linea in cui si è verificato l'errore. RESUME NEXT riporta l'esecuzione all'istruzione seguente quella contenente l'errore; RESUME con un numero di linea salterà alla linea specificata, riprendendo da lì.

RETURN RETURN

Questa istruzione è sempre usata con l'istruzione GOSUB. Quando il programma incontra un'istruzione RETURN, va all'istruzione immediatamente seguente l'ultimo comando GOSUB. Se non sono stati eseguiti precedentemente GOSUB, il computer si presenterà con un messaggio di errore del tipo RETURN WITHOUT GOSUB ERROR.

SCALE SCALE <1/0>

La scala dei modi alta risoluzione e multicolore può essere modificata con il comando SCALE.

ESEMPI:

SCALE 1

Cambia la scala. Le coordinate vanno da 0 a 1023 sia sulla X che sulla Y al posto della normale scala di valori che è:
Modo multicolore ... X da 0 a 159, Y da 0 a 199
Modo alta risoluzione ... X da 0 a 319, Y da 0 a 199
La scala può essere riportata alla normalità eseguendo SCALE 0.

SCNCLR SCNCLR

Cancella lo schermo corrente, sia grafico, testuale o entrambi (nel caso di schermo misto).

SOUND SOUND Numero voce, controllo frequenza, durata.

Questa istruzione produce un suono usando una delle tre voci con un controllo di frequenza che va da 0 a 1023 per una durata da 0 a 65535 sessantesimi di secondo.

V	Voce
1	Voce 1 (tono)
2	Voce 2 (tono)
3	Voce 2 (rumore bianco)

Se è richiesto un suono dalla voce N prima che il suono precedente sia terminato, il BASIC aspetta che il primo sia completato. Il suono con durata 0 è un caso speciale. Fa in modo che il BASIC interrompa il suono corrente immediatamente, qualunque sia la durata. La tavola delle note musicali nell'appendice riporta i valori del controllo di frequenza che corrispondono alle note musicali.

ESEMPIO:

SOUND 2,800,360

Genera una nota usando la voce 2 con frequenza 800 per un minuto.

SSHAPE/GSHAPE

SSHAPE e GSHAPE sono usati per salvare e richiamare aree rettangolari dalla memoria multicolore o ad alta risoluzione usando variabili stringa BASIC. Il comando per salvare un'area è:

SSHAPE variabile stringa, a1, b1 [, a2, b2]

Variabile stringa	Nome della stringa in cui salvare i dati.
a1, b1	Coordinate di un vertice in scala.
a2, b2	Coordinate del vertice opposto ad a1, b1 (normalmente è il PC).

Poiché la lunghezza delle stringhe BASIC è limitata a 255 caratteri, le dimensioni dell'area che potete salvare è limitata. La dimensione della stringa richiesta può essere calcolata usando una delle seguenti formule non in scala

L (MCM) = INT ((ABS(a1-a2)+1)/4+.99)*(ABS(b1-b2)+1)+4
L(HR)=INT((ABS(a1-a2)+1)/4+.99)*(ABS(b1-b2)+1)+4

L'area è salvata riga per riga. Gli ultimi 4 bytes della stringa contengono le lunghezze di riga e colonna meno 1 nel formato byte alto/byte basso (se in scala modificata dividere la lunghezza X per 3.2 e la lunghezza Y per 5.12).
Il comando per visualizzare un'area salvata su una qualsiasi area dello schermo è:

GSHAPE stringa [,[a, b] [modo]]

Stringa	Contiene l'area da disegnare
a, b	Coordinate del vertice in alto a sinistra. Normalm. è il PC
Modo	Modo di rappresentazione. 0=Piazza la figura così com'è. 1=Piazza la figura in campo inverso. 2=Esegue l'OR tra la figura e l'area. 3=Esegue l'AND tra la figura e l'area. 4=Esegue l'XOR tra la figura e l'area.

ESEMPI:

SSHAPE "CIAO", 0,0	Salva l'area di schermo dal vertice più in alto a sinistra a dove è posizionato il cursore, col nome CIAO.
GSHAPE "CIAO",,,1	Mostra la figura CIAO con i colori di sfondo e di primo piano invertiti, con il vertice in alto a sinistra della figura coincidente col cursore.

STOP STOP

Questa istruzione ferma il programma e provoca la stampa del messaggio BREAK IN LINE N, dove "N" è il numero di linea che contiene STOP. Il programma può essere fatto ripartire dall'istruzione seguente la STOP usando il comando CONT. L'istruzione STOP è normalmente usata per correggere un programma.

SYS SYS indirizzo

La parola SYS è seguita da un numero decimale o da una variabile numerica che va da 0 a 65535. Il programma comincia l'esecuzione del programma in linguaggio macchina a partire dalla locazione specificata. Questo comando è simile alla funzione USR, ma non passa un parametro. Vedere il COMMODORE 16 PROGRAMMER'S REFERENCE GUIDE per informazioni sui programmi in linguaggio macchina.

TRAP TRAP [Numero linea]

Quando attivato, TRAP intercetta tutte le condizioni di errore incluso il tasto RUN/STOP, eccetto gli UNDEF'D STATEMENT ERROR. Nel caso che avvenga un qualsiasi errore di esecuzione, il flag di error è attivo, e l'esecuzione è trasferita al numero di linea specificato nell'istruzione TRAP. Il numero di linea nel quale avviene l'errore può essere trovato usando la variabile di sistema EL. Il tipo di errore è contenuto nella variabile di sistema ER. La funzione stringa ERR$ (ER) fornisce il messaggio di errore corrispondente ad ogni condizione (ER) di errore.

> NOTA: Un errore in una routine di TRAP non può essere intercettato. L'istruzione RESUME può essere usata per riprendere l'esecuzione del programma. TRAP senza numero di linea termina l'intercettazione degli errori.

TRON TRON

TRON viene usato nella correzione dei programmi. Questa istruzione inizia il modo di TRACE. Quando siete nel modo TRACE, viene stampato il numero di linea di ogni istruzione eseguita.

TROFF TROFF

Questa istruzione disabilita il modo TRACE.

VOL VOL Livello volume.

Pone il livello di volume al valore specificato, per i comandi SOUND. Il livello di volume può essere scelto tra 0 e 8, dove 8 è il volume massimo, 0 è spento. L'istruzione VOL influenza entrambe le voci.

WAIT WAIT Indirizzo, valore 1 [,valore 2]

L'istruzione WAIT è usata per fermare il programma quando il contenuto di una locazione di memoria cambia in un modo specifico. L'indirizzo deve essere compreso tra 0 e 65535. Il valore 1 e il valore 2 devono essere compresi tra 0 e 255.
Viene innanzitutto eseguito un OR esclusivo tra il contenuto della locazione di memoria e il valore 2 (se presente), e poi un AND logico col valore 1. Finché il risultato è 0 il programma continua a controllare la locazione di memoria. Quando il risultato è diverso da 0 il programma riprende il suo corso.

Alcune Informazioni sulle Istruzioni Grafiche

Ci sono alcuni concetti che si applicano a tutte le istruzioni grafiche. Il primo è il concetto di Pixel Cursor (PC). Il PC è simile al cursore del modo testo; è la posizione dove sarà disegnato il prossimo punto. Diversamente dal cursore di testo, il PC è invisibile. Tutti i comandi di disegno usano il PC. In aggiunta il comando LOCATE vi permette di spostare il PC senza disegnare alcunché. Dovunque vogliate usare le coordinate X, Y in un comando di DRAW, potete usare invece coordinate RELATIVE. Le coordinate relative sono basate sul valore corrente del PC. Per usare coordinate relative, piazzare un "+" o un "-" prima delle coordinate. Un segno "+" prima del valore della X sposta il PC a destra. Un segno "-" prima del valore della X sposta il PC a sinistra. Messi davanti alla Y, spostano il PC rispettivamente in su e in giù.

ESEMPIO:

LOCATE + 100, - 25	Muove il PC a destra di 100 punti e su di 25.
DRAW 1, +10, +10 to 100, 100	Disegna una linea 10 punti a destra e 10 sotto il valore del PC fino al punto assoluto 100, 100.

Si può anche specificare una distanza ed un angolo relativo dal PC, separando due parametri con ";".

ESEMPIO:

LOCATE 50:45	Muove il PC dalla sua posizione alla distanza di 50 punti ad un angolo di 45 gradi.

FUNZIONI
Funzioni numeriche

Le funzioni numeriche sono classificate come tali poiché ritornano in numero.
Le funzioni numeriche seguono la forma:
FUNZIONE (argomento)
Dove l'argomento può essere un valore numerico, una variabile o una stringa.

ABS (X)
La funzione valore assoluto ritorna il valore positivo dell'argomento X.

ASC (X$)
Questa funzione ritorna il codice ASCII (numero) del primo carattere di X$.

ATN (X)
L'arcotangente X ritorna l'angolo la cui tangente è X, misurato in radianti.

COS (X)
Ritorna il valore del coseno di X, dove X è un angolo misurato in radianti.

DEC (Stringa esadecimale)
Ritorna il valore decimale della stringa esadecimale (tra 0 e FFFF).

ESEMPIO:

N=DEC("F4")

EXP (X)
Ritorna il valore della costante matematica e (2.71828183) elevata alla X.

FNxx (X)
Ritorna il valore della funzione xx definita dall'utente creata in un'istruzione DEF FNxx(X).

INSTR (stringa 1, stringa 2 [posizione di partenza])
Ritorna la posizione della stringa 2 nella stringa 1 dall'inizio o dopo la posizione di partenza. La posizione di partenza normale è l'inizio della stringa 1. Se non è trovata la stringa 2, viene ritornato un valore 0.

ESEMPIO:

PRINT INSTR ("IL GATTO NEL CAPPELLO", "GATTO") il risultato è 4, poiché GATTO parte dal quarto carattere della stringa.

INT (X)
Ritorna la porzione intera di X, con tutte le cifre decimali a destra del punto rimosse. Il risultato è sempre minore o uguale a X. Allo stesso modo ogni numero negativo con decimali diventa il numero intero minore del suo valore corrente. Se la funzione deve essere usata per arrotondare per eccesso o per difetto, la forma è: INT (X +/- .5)

ESEMPI:

X=INT(X*100+.5)/100	Arrotonda all'intero più vicino.
X=INT(-4.5)	Il risultato è -5.

JOY (n).
Quando n=1, la funzione restituisce la posizione del joystick 1, quando n=2 la posizione del joystick 2.
Ogni valore maggiore di 128 significa che è premuto anche il tasto di FIRE. La direzione è indicata come segue:

ESEMPIO:

JOY(2)=135	Il Joystick 2 spara sulla sinistra.

LOG (X)
Ritorna il logaritmo naturale di X. Il logaritmo naturale è il logaritmo in base e (vedere EXP (X)). Per convertire il logaritmo in base 10 dividere per il logaritmo di 10.

PEEK (X)
Questa funzione fornisce il contenuto della locazione di memoria X compresa tra 0 e 65535, riportando un risultato da 0 a 255. Questa è spesso usato in congiunzione con l'istruzione POKE.

RCLR (N)
Ritorna il colore assegnato alla sorgente N, compresa tra 0 e 4 (0=sfondo, 1=Primo Piano, 2=multicolore 1, 3=multicolore 2, 4=bordo).

RDOT (N)
Ritorna informazioni riguardo alla posizione corrente del PC. Se N=0, ritorna la coordinata X, 1 la coordinata Y, 2 la sorgente colore.

RGR (X)
Ritorna il modo grafico corrente, X è un argomento insignificante.

RLUM (N)
Ritorna il livello corrente di luminosità assegnato alla sorgente colore N.

RND (X)
Questa funzione da un numero casuale compreso tra 0 e 1, molto utile nei giochi, per simulare un lancio di dadi o altri elementi variabili, ed è anche usata in alcune applicazioni statistiche. Il primo numero casuale può essere generato con la formula RND (-TI), per iniziare il tutto da un punto diverso ogni volta. In seguito il numero X può essere 1, o un qualsiasi numero positivo. Se X è 0, viene ritornato l'ultimo numero casuale generato. Un valore negativo di X inizializza una serie di numeri casuali con X. L'uso dello stesso valore negativo di X da come risultato la stessa serie di numeri casuali. Un valore positivo di X darà come risultato un numero casuale basato sul precedente.
Per simulare il lancio di un dado, si usa la formula INT (RND(1)*6+1). Prima il numero casuale tra 0 e 1 (esclusi) viene moltiplicato per 6, cosa che espande il campo di variabilità tra 0 e 6 (esclusi), poi viene sommato 1, che darà un numero tra 1 e 7 (esclusi). La funzione INT tronca tutte le cifre decimali, lasciando come risultato una cifra da 1 a 6 (compresi).

Per simulare due dadi, sommare due dei numeri ottenuti con la formula precedente.

ESEMPIO:

100 X=INT(RND(1)*6)+INT(RND(1)*6)+2	Simula due dadi.
100 X=INT(RND(1)*1000)+1	Numero da 1 a 1000
100 X=INT(RND(1)*150)+100	Numero da 100 a 249

SGN(X)
Questa funzione ritorna il segno dell'argomento. Il risultato è +1 se positivo 0 se 0, -1 se negativo.

SIN(X)
Questa è la funzione trigonometrica del seno. Il risultato è il seno di X, dove X è un angolo in radianti.

SQR(X)
Questa funzione dà la radice quadrata di X, dove X è un numero maggiore o uguale a zero. Se X è negativo, risulta un ILLEGAL QUANTITY ERROR.

TAN(X)
Questa fornisce la tangente di X, dove X è un angolo in radianti.

USR(X)
Quando questa funzione è usata, il programma salta al programma in linguaggio macchina, la cui locazione di partenza è contenuta nelle locazioni di memoria 1281 e 1282. Il parametro X è passato al programma in linguaggio macchina nel FLOATING POINT ACCUMULATOR. Un altro numero viene restituito al BASIC attraverso la variabile chiamante. In altre parole, questo vi permette di scambiare una variabile tra il linguaggio macchina e il BASIC. Vedere il COMMODORE 16 PROGRAMMER'S REFERENCE GUIDE per maggiori dettagli sulla programmazione in linguaggio macchina.

VAL (X$)
Questa funzione converte la stringa X$ in un numero ed è essenzialmente la funzione inversa della STR$. La stringa è esaminata da sinistra a destra, alla ricerca dei caratteri in formato numerico riconoscibile tra tutti quelli presenti nella stringa. Se il Commodore 16 trova caratteri illegali, solo la porzione di stringa fino a quel punto è convertita.

ESEMPI:

```
10 X=VAL("123.456")          X=123.456
10 X=VAL("3E03")             X=3000 (La E è intesa come esponente).
10 X=VAL("12A13B")           X=12
10 X=VAL("RYUO17*")          X=0
10 X=VAL("-1.23.23.23")      X=-1.23
```

Funzioni stringa

Le funzioni stringa si differenziano da quelle numeriche in quanto ritornano caratteri, grafici o numeri, da una stringa (definita da "") al posto di un numero.

CHR$(X)
Questa funzione ritorna un carattere stringa il cui codice ASCII è X.

ERR$(X)
Questa funzione ritorna una stringa che descrive la condizione di errore X (Vedere TRAP).

HEX$(X)
Ritorna una stringa di quattro caratteri contenente la rappresentazione esadecimale del numero X. (0<X<65535)

LEFT$(X$,X)
Questa ritorna una stringa contenente gli X caratteri più a sinistra di X$.

LEN(X$)
Ritorna il numero di caratteri (inclusi spazi ed altri simboli) della stringa X$.

MID$(X$,S,X)
Questa funzione ritorna una stringa contenente X caratteri, partendo dal carattere S-esimo di X$. MID$ può anche essere usata sulla parte sinistra delle funzioni di assegnamento come una pseudo variabile.
Questa funzione sostituisce nella stringa, X$, a partire dal S-esimo carattere e per X caratteri la stringa indicata dopo il segno "=". La lunghezza normalmente è quella delle variabili stringa, e verrà dato un messaggio di errore se la posizione di partenza più la lunghezza saranno maggiori della lunghezza della variabile stringa.

ESEMPI:

```
10 A$="L'ULTIMO CIAO!"
20 PRINT A$                       Stampa "L'ULTIMO CIAO!"
30 MID$(A$, 4,5)= "NICO"
40 PRINT A$                       Stampa "L'UNICO    CIAO!"
```

RIGHT$ (X$,X)
Ritorna gli X-caratteri più a destra di X$.

STR$(X)
Ritorna una stringa che è identica alla versione stampata di X.

ESEMPIO:

```
A$=STR$(143)        A$="143"
```

Altre Funzioni

FRE(X)
Questa funzione ritorna il numero di bytes inutilizzati e quindi disponibili in memoria. X è un argomento insignificante.

POS(X)
Questa funzione ritorna il numero della colonna tra 0 e 79 dove la prossima istruzione PRINT inizierà a stampare sullo schermo.

SPC(X)
È usata nell'istruzione PRINT per saltare X-spazi. X può avere un valore da 0 a 255.
Va con l'istruzione PRINT. Il prossimo argomento è da stampare nella colonna X. X può avere un valore da 0 a 255.

TAB(X)
È usata nell'istruzione PRINT. La stampa del prossimo dato verrà nella colonna X. X può avere un valore da 0 a 255.

π
Il simbolo pigreco, quando è usato in un'equazione dà come valore 3.14159265

VARIABILI E OPERATORI
Variabili

Il vostro Commodore 16 usa tre tipi di variabili in BASIC. Ci sono: variabili numeriche in virgola mobile, variabili numeriche intere e variabili stringa (alfanumeriche).
Le **variabili numeriche** in virgola mobile possono avere un qualsiasi valore da 1E-38 a 1E+38, con più di nove cifre significative. Quando un numero diventa più grande delle nove cifre rappresentabili, di dieci in dieci, il vostro computer lo mostra con la notazione scientifica, col numero normalizzato ad una cifra, e con le otto cifre decimali seguite dalla lettera E e la potenza di dieci per la quale il numero è moltiplicato. Per es., il numero 12345678901 è rappresentato come 1.23456789E+10.
Le **variabili intere** possono essere usate quando il numero è compreso tra +32768 e -32768, e non è frazionario. Un intero è un numero come 5, -10,81.
Gli interi occupano meno spazio delle variabili in virgola mobile, specialmente quando sono usate in una matrice.
Le **variabili stringa** sono usate per dati alfanumerici, e possono contenere numeri, lettere ed ogni altro carattere che il vostro Commodore 16 possa produrre. Un esempio di variabile stringa è "COMMODORE 16".

Nomi di variabili

I nomi di variabili possono consistere in una sola lettera, una lettera seguita da un numero, o in due lettere. I nomi di variabili possono essere più lunghi di due caratteri, ma solo i primi due sono significativi.
Una variabile intera è specificata usando il segno "%" dopo il nome della variabile; le variabili stringa con '$' dopo il nome.

ESEMPI:

Nome di variabile numerica	A,A5,BZ
Nome di variabile intera	A%,A5%,BZ%
Nome di variabile stringa	A$,A5$,BZ$

Le matrici sono liste di variabili con lo stesso nome, che usano uno o più numeri extra per specificare un elemento della matrice. Le matrici sono definite usando l'istruzione DIM, e possono essere reali, intere o di stringa. Il nome della matrice è seguito da parentesi che racchiudono il numero delli variabili della lista.

ESEMPI:

A(7),BZ%(11),A$(87)

Le matrici possono avere più di una dimensione. Le matrici bidimensionali possono essere viste come formate da righe e colonne, col primo indice che specifica la riga ed il secondo che specifica la colonna.

ESEMPI:

A(7,2),BZ%(2,3,4)Z$(3,2)

Nomi di variabili riservate

Ci sono sette nomi di variabili il cui uso è riservato al Commodore 16 e non possono essere usate per altri scopi. Sono le variabili D$,DS$,ER,EL,ST,TI,TI$. Non potete usare nemmeno le parole-chiave TO e IF o qualsiasi nome che contenga parole-chiave, come SRUN, RNEW, XLOAD come nomi di variabili. ST è una variabile di stato per l'ingresso e l'uscita (eccetto le normali operazioni di schermo e tastiera). Il valore di ST dipende dal risultato dell'ultima operazione di ingresso/uscita. Una spiegazione più dettagliata di ST si trova nel COMMODDRE 16 PROGRAMMER'S REFERENGE GUIDE, ma in generale se il valore di ST è 0, l'operazione ha avuto successo.
TI e TI$ sono variabili legate all'orologio in tempo reale esistente nel vostro Commodore 16. L'orologio di sistema è incrementato ogni sessantesimo di secondo. Parte da zero quando il vostro Commodore 16 viene acceso, ed è resettato solo cambiando valore a TI$. La variabile TI vi fornisce il valore corrente dell'orologio in sessantesimi di secondo.
TI$ è una stringa che legge il valore dell'orologio in tempo reale come un orologio a 24 ore. I primi due caratteri di TI$ contengono le ore, il terzo e quarto i minuti, il quinto e il sesto sono i secondi. A questa variabile può essere assegnato qualsiasi valore, a patto che tutti i caratteri siano numeri, e sarà automaticamente incrementata come un orologio.

ESEMPIO:

TI$="101530"   Regola l'orologio alle 10:15 e 30 secondi.

Il valore dell'orologio viene perso quando il Commodore viene spento.
La variabile DS legge il canale di comando del disco, e ritorna lo stato attuale del drive. Per avere queste informazioni in parole, fare PRINT DS$. Queste variabili di stato sono usate dopo un'operazione su disco, come DLOAD o DSAVE per scoprire perché la luce rossa di errore sul drive sta lampeggiando.
ER, EL ed ERR$ sono variabili usate nelle routine di recupero dell'errore (vedi TRAP). Sono utili solo se usate in programma. ER ritorna l'ultimo errore avvenuto da quando è stato fatto partire il programma. EL è il numero di linea dove è avvenuto l'errore. ERR$ è la funzione che permette al vostro programma di stampare uno dei messaggi di errore. PRINT ERR$(ER) stampa il corrispondente messaggio di errore.

OPERATORI BASIC

Gli OPERATORI ARITMETICI comprendono i seguenti segni:

'+'	addizione
'-'	sottrazione
'*	moltiplicazione
'/'	divisione
'↑'	elevamento a potenza

In una linea contenente più di un operatore le varie operazioni vengono eseguite secondo l'ordine classico della matematica, cioè verrà effettuata prima l'elevazione, poi la moltiplicazione e la divisione, e per ultime l'addizione e sottrazione. Se due operazioni hanno la stessa priorità, vengono calcolate da sinistra a destra. Se volete che queste operazioni avvengano in un diverso ordine, il Commodore 16 vi permette di dare una maggiore priorità ad un calcolo racchiudendolo tra parentesi. Le operazioni poste tra parentesi saranno calcolate prima di ogni altra. Dovète accertarvi che le vostre equazioni abbiano lo stesso numero di parentesi aperte e chiuse, altrimenti otterrete un messaggio di SYNTAX ERROR.
Ci sono altri operatori di eguaglianza e diseguaglianza chiamati OPERATORI DI RELAZIONE. Gli operatori aritmetici hanno sempre priorità sugli operatori di relazione.

=	uguale
<	minore di
>	maggiore di
<= o =<	minore o uguale a
>= o =>	maggiore o uguale a
<> o ><	diverso da

Infine ci sono tre operatori logici, con priorità più bassa di tutti gli altri: AND, OR, NOT.
Questi sono usati spesso per collegare formule multiple nelle istruzioni IF...THEN.
Quando sono usati con operatori aritmetici sono valutati per ultimi.

ESEMPI:

IF A=B AND C=D THEN 100	Richiede che entrambi le condizioni siano vere per eseguire il THEN.
IF A=B OR C=D THEN 100	Richiede che una delle condizioni sia vera.
A=5:B=4:PRINT A=B	Stampa il valore 0.
A=5:B=4:PRINT A>B	Stampa il valore -1.
PRINT 123 AND 15:PRINT 5 OR 7	Stampa 11 e 7

Abbreviazione Basic 3.5

PAROLA CHIAVE	ABBREVIAZIONE			TIPO
ABS	a	SHIFT	B	funzione-numerica
ASC	a	SHIFT	S	funzione-numerica
ATN	a	SHIFT	T	funzione-numerica
AUTO	a	SHIFT	U	comando
BACKUP	b	SHIFT	A	comando
BOX	b	SHIFT	O	istruzione
CHAR	ch	SHIFT	A	istruzione
CHR$	c	SHIFT	H	funzione-stringa
CIRCLE	c	SHIFT	I	istruzione
CLOSE	cl	SHIFT	O	istruzione
CLR	c	SHIFT	L	istruzione
CMD	c	SHIFT	M	istruzione
COLLECT	col	SHIFT	L	comando
COLOR	co	SHIFT	L	istruzione
CONT	c	SHIFT	O	comando
COPY	co	SHIFT	P	comando
COS		nessuna		funzione-numerica
DATA	d	SHIFT	A	istruzione
DEC		nessuna		funzione-numerica
DEFFN	d	SHIFT	E	istruzione
DELETE	de	SHIFT	L	comando
DIM	d	SHIFT	I	istruzione
DIRECTORY	di	SHIFT	R	comando
DLOAD	d	SHIFT	L	comando
DO		nessuna		istruzione
DRAW	d	SHIFT	R	istruzione
DSAVE	d	SHIFT	S	comando
END	e	SHIFT	N	istruzione
ERR$	e	SHIFT	R	funzione-stringa
EXP	e	SHIFT	X	funzione-numerica
FOR	f	SHIFT	O	istruzione
FRE	f	SHIFT	R	funzione-numerica
GET	g	SHIFT	E	istruzione
GETKEY	getk	SHIFT	E	istruzione
GET		nessuna		istruzione
GOSUB	go	SHIFT	S	istruzione
GOTO	g	SHIFT	O	istruzione
GRAPHIC	g	SHIFT	R	istruzione
GSHAPE	g	SHIFT	S	istruzione
HEADER	he	SHIFT	A	comando
HEX$	h	SHIFT	E	funzione-stringa

IF...GOTO		nessuna		istruzione
IF...THEN...ELSE		nessuna		istruzione
INPUT		nessuna		istruzione
INPUT	i	SHIFT	N	istruzione
INSTR	in	SHIFT	S	funzione-numerica
INT		nessuna		funzione-numerica
JOY	j	SHIFT	O	funzione-numerica
KEY	k	SHIFT	E	comando
LEFT$	le	SHIFT	F	funzione-stringa
LEN		nessuna		funzione-numerica
LET	l	SHIFT	E	istruzione
LIST	l	SHIFT	I	comando
LOAD	l	SHIFT	O	comando
LOCATE	lo	SHIFT	C	istruzione
LOG		nessuna		funzione-nnumerica
LOOP	lo	SHIFT	O	istruzione
MID$	m	SHIFT	I	funzione-stringa
MONITOR	m	SHIFT	O	istruzione
NEW		nessuna		comando
NEXT	n	SHIFT	E	istruzione
ON...GOSUB	on...go	SHIFT	S	istruzione
ON...GOTO	on...g	SHIFT	O	istruzione
OPEN	o	SHIFT	P	istruzione
PAINT	p	SHIFT	A	istruzione
PEEK	p	SHIFT	E	funzione-numerica
POKE	p	SHIFT	O	istruzione
POS		nessuna		funzione-numerica
PRINT	?			istruzione
PRINT	p	SHIFT	R	istruzione
PRINT USING	?us	SHIFT	I	istruzione
PUDEFF	p	SHIFT	U	istruzione
RCLR	r	SHIFT	C	funzione-numerica
RDOT	r	SHIFT	D	funzione-numerica
READ	r	SHIFT	E	istruzione
REM		nessuna		istruzione
RENAME	re	SHIFT	N	comando
RENUMBER	ren	SHIFT	U	comando
RESTORE	re	SHIFT	S	istruzione
RESUME	res	SHIFT	U	istruzione
RETURN	re	SHIFT	T	istruzione
RGR	r	SHIFT	G	funzione-numerica

RIGHT$	r	SHIFT	I	funzione-stringa
RLUM	r	SHIFT	L	funzione-numerica
RND	r	SHIFT	N	funzione-numerica
RUN	r	SHIFT	U	comando
SAVE	s	SHIFT	A	comando
SCALE	sc	SHIFT	A	istruzione
SCNCLR	s	SHIFT	C	istruzione
SCRATCH	sc	SHIFT	R	comando
SGN	s	SHIFT	G	funzione-numerica
SIN	s	SHIFT	I	funzione-numerica
SOUND	s	SHIFT	O	istruzione
SPC	s	SHIFT	P	funzione-speciale
SQR	s	SHIFT	Q	funzione-numerica
SSHAPE	s	SHIFT	S	istruzione
STatus		nessuna		riservato-variabile-numerica
STOP	s	SHIFT	T	istruzione
STR$	st	SHIFT	R	funzione-stringa
SYS	s	SHIFT	Y	istruzione
TAB	t	SHIFT	A	funzione-speciale
TAN		nessuna		funzione-numerica
TI		nessuna		riservato-variabile-numerica
TI$		nessuna		riservato-variabile di stringa
TRAP	t	SHIFT	R	istruzione
TROFF	tro	SHIFT	F	istruzione
TRON	tr	SHIFT	O	istruzione
UNTIL	u	SHIFT	N	istruzione
USR	u	SHIFT	S	funzione-speciale
VAL		nessuna		funzione-numerica
VERIFY	v	SHIFT	E	comando
VOL	v	SHIFT	O	istruzione
WAIT	w	SHIFT	A	istruzione
WHILE	w	SHIFT	H	istruzione

APPENDICI

- Messaggi di errore
- Messaggi di errore del disco
- Calcolo di funzione matematiche
- Tabella delle note musicali
- Codici dei caratteri di schermo e ascii
- Lista dei libri

APPENDICE A

Messaggi d'Errore

Questi messaggi d'errore vengono stampati dal BASIC. Si potrà anche visualizzare i messaggi tramite l'uso della funzione ERR$ (). Per quanto riguarda questa funzione il numero d'errore si riferisce solo al numero assegnato all'errore.

	ERRORE	NOME DELL'ERRORE
1	**TOO MANY FILES (TROPPI FILE)**	Il limite dei file aperti contemporaneamente è 10.
2	**FILE OPEN (FILE APERTO)**	È stato fatto un tentativo di aprire un file utilizzando il numero di un file già aperto.
3	**FILE NOT OPEN (FILE NON APERTO)**	Il numero di file specificato in una istruzione I/O deve essere aperto prima di utilizzarlo.
4	**FILE NOT FOUND (FILE NON TROVATO)**	O non esiste nessun file con quel nome (disco) o è stato letto un indicatore di fine nastro (nastro).
5	**DEVICE NOT PRESENT (DISPOSITIVO NON PRESENTE)**	Il dispositivo I/O richiesto non è disponibile.
6	**NOT INPUT FILE (FILE NON DI INPUT)**	È stato fatto un tentativo per ottenere o introdurre dati di un file che è stato specificato di solo output.
7	**NOT OUTPUT FILE (FILE NON DI OUTPUT)**	È stato fatto un tentativo di trasmettere dati ad un file che è stato specificato di solo input.
8	**MISSING FILE NAME (MANCA IL NOME DEL FILE)**	Le istruzioni OPEN, LOAD, o SAVE trasmesse all'unità disco richiedono generalmente un nome di file.
9	**ILLEGAL DEVICE NUMBER (NUMERO DI DISPOSITIVO ILLEGALE)**	È stato fatto un tentativo di utilizzare non correttamente un dispositivo (istruzione SAVE applicata allo schermo, ecc.).

10	**NEXT WITHOUT FOR (NEXT SENZA FOR)**	O i loop non sono nidificati correttamente, o c'è un nome di variabile in una istruzione NEXT che non corrisponde a quello contenuto in un'istruzione FOR.
11	**SYNTAX (SINTASSI)**	Una istruzione non è riconoscibile dal BASIC. Questo può accadere a causa della mancanza di parentesi, di una parola chiave mal formulata, ecc.
12	**RETURN WITHOUT GOSUB (RETURN SENZA GOSUB)**	Un'istruzione RETURN è stata incontrata quando nessuna istruzione GOSUB era attiva.
13	**OUT OF DATA (DATI ESAURITI)**	È stata incontrata un'istruzione READ (leggere) senza che nessun dato sia rimasto non-letto.
14	**ILLEGAL QUANTITY (QUANTITÀ ILLEGALE)**	Un numero usato come argomento di una funzione o di una istruzione è al di fuori della gamma permessa.
15	**OVERFLOW**	Il risultato di un calcolo è maggiore del numero massimo permesso (1.701411833E+38).
16	**OUT OF MEMORY (MEMORIA ESAURITA)**	O non c'è più spazio per il programma e le variabili di programma, o ci sono troppe istruzioni DO, FOR o GOSUB in funzione.
17	**UNDEF'D STATEMENT (ISTRUZIONE NON DEFINITA)**	Il numero di riga voluto non esiste nel programma.
18	**BAD SUBSCRIPT (INDICE NON VALIDO)**	Il programma ha cercato di indirizzare un elemento di una matrice fuori della gamma specificata dalla istruzione DIM.
19	**REDIM'D ARRAY (MATRICE RIDIMENSIONATA)**	Una matrice può essere dimensionata solo una volta. Se una matrice viene indirizzata prima di essere dimensionata, viene effettuato un dimensionamento automatico (fino a 10).
20	**DIVISION BY ZERO (DIVISIONE PER ZERO)**	La divisione per zero non è ammessa.
21	**ILLEGAL DIRECT (INDIRIZZO ILLEGALE)**	INPUT o GET sono istruzioni permesse solo all'interno di un programma.
22	**TYPE MISMATCH (ERRORE DI BATTITURA)**	Questo avviene quando un numero viene usato al posto di una stringa o viceversa.
23	**STRING TOO LONG (STRINGA TROPPO LUNGA)**	Una stringa può contenere fino a 255 caratteri.

24	**FILE DATA (DATI DEL FILE)**	Dati non corretti letti da un file su nastro.
25	**FORMULA TOO COMPLEX (FORMULA TROPPO COMPLESSA)**	Semplificare l'espressione (dividere in due parti oppure utilizzare meno parentesi)
26	**CAN'T CONTINUE (NON È POSSIBILE CONTINUARE)**	Il comando CONT non funziona se il programma non è stato lanciato, se c'è un errore o se è stata modificata una riga.
27	**UNDEF'D FUNCTION (FUNZIONE NON DEFINITA)**	Si è fatto riferimento ad una funzione definita dall'utente che non era mai stata definita.
28	**VERIFY (VERIFICA)**	Il programma su nastro o su disco non corrisponde al programma in memoria.
29	**LOAD (CARICAMENTO)**	C'è stato un problema nel caricamento. Riprovare.
30	**BREAK (INTERRUZIONE)**	Si è battuto il tasto STOP per interrompere l'esecuzione del programma.
31	**CAN'T RESUME (RECUPERO IMPOSSIBILE)**	Si è incontrata un'istruzione RESUME senza un'istruzione TRAP attiva.
32	**LOOP NOT FOUND (LOOP NON TROVATO)**	Il programma ha incontrato una istruzione DO e non può trovare il LOOP corrispondente.
33	**LOOP WITHOUT DO (LOOP senza DO)**	Si è incontrato un LOOP senza un'istruzione DO attiva.
34	**DIRECT MODE ONLY (SOLO MODALITÀ DIRETTA)**	Questo comando è permesso solo in modalità diretta, non da un programma.
35	**NO GRAPHICS AREA (AREA NON GRAFICA)**	Si è incontrato un comando (DRAW, BOX, ecc.) per creare grafici prima che fosse eseguito un comando GRAPHIC.
36	**BAD DISK (DISCO DIFETTOSO)**	Si è tentato di riformattare un dischetto non formattato o difettoso con il metodo di formattazione veloce (senza ID).

APPENDICE B

DESCRIZIONE DEI MESSAGGI D'ERRORE DOS

Questi messaggi d'errore vengono ritornati attraverso le variabili riservate DS e DS$.

NOTA: I numeri di messaggio d'errore inferiori a 20 dovrebbero venire ignorati eccetto 01., che informa sul numero di file cancellati con il comando SCRATCH.

20	**READ ERROR** (ERRORE DI LETTURA) (intestazione del blocco non trovata)	Il controllore del disco non può allocare l'intestazione del blocco di dati richiesto. Ciò può essere provocato da un numero di settore illegale, o dal fatto che l'intestazione è stata distrutta.
21	**READ ERROR** (ERRORE DI LETTURA) (carattere non sincronizzato)	Il controllore del disco non può rilevare un indicatore di sincronizzazione sulla traccia desiderata. Questo può essere provocato da un disallineamento della testina di lettura/scrittura, dalla mancanza del dischetto, da un dischetto non formattato o inserito non correttamente. Può anche indicare un guasto hardware.
22	**READ ERROR** (ERRORE DI LETTURA) (blocco dati non presente)	Al controllore del disco è stato richiesto di leggere o verificare un blocco di dati che non è stato scritto correttamente. Questo messaggio d'errore si presenta in congiunzione con i comandi BLOCK ed indica una richiesta di traccia e/o settore illegale.
23	**READ ERROR** (ERRORE DI LETTURA) (errore di checksum nel blocco di dati)	Questo messaggio d'errore indica che c'è un errore in uno o più d'uno dei byte dei dati. I dati sono stati letti nella memoria DOS, ma il checksum sui dati è in errore. Questo messaggio può anche indicare problemi di messa a terra.
24	**READ ERROR** (ERRORE DI LETTURA) (errore di decodificazione del byte)	I dati o l'intestazione sono stati letti nella memoria DOS, ma si è creato un errore hardware a causa di una configurazione di bit non valida in un byte di dati. Questo messaggio può anche indicare problemi di messa a terra.

25	**WRITE ERROR** (ERRORE DI SCRITTURA) (errore di verifica di scrittura)	Questo messaggio viene generato se il controllore rileva un'incongruenza tra i dati scritti e i dati della memoria DOS
26	**WRITE PROTECT ON** (PROTEZIONE IN SCRITTURA SULL'ON)	Questo messaggio viene generato quando è stato richiesto al controllore di scrivere un blocco di dati mentre l'interruttore di protezione di scrittura è sull'on. Di solito questo viene causato utilizzando un dischetto protetto in scrittura.
27	**READ ERROR** (ERRORE DI LETTURA) (errore di checksum nella intestazione)	Il controllore ha rilevato un errore nell'intestazione del blocco di dati richiesto. Il blocco non è stato letto nella memoria DOS. Questo messaggio può anche indicare problemi di messa a terra.
28	**WRITE ERROR** (ERRORE DI SCRITTURA) (blocco di dati lungo)	Il controllore tenta di rilevare l'indicatore di sincronizzazione dell'intestazione seguente dopo aver scritto un blocco di dati. Se l'indicatore di sincronizzazione non appare entro un tempo predeterminato, viene generato il messaggio d'errore. L'errore viene causato da una formattazione non corretta del dischetto (i dati si estendono nel blocco seguente) o da un guasto hardware.
29	**DISK ID MISMATCH** (ERRORE DI IDENTIFICAZIONE DEL DISCO)	Questo messaggio viene generato quando è stato richiesto al controllore di accedere a un dischetto che non è stato inizializzato. Il messaggio può anche presentarsi se un dischetto ha un'intestazione non corretta.
30	**SYNTAX ERROR** (ERRORE DI SINTASSI) (sintassi generale)	Il DOS non può interpretare il comando inviato al canale di comando. Solitamente questo errore viene provocato da un numero illegale di nomi di file, o dal fatto che le configurazioni vengono usate illegalmente. Per esempio, due nomi di file possono apparire sul lato sinistro del comando COPY.
31	**SYNTAX ERROR** (ERRORE DI SINTASSI) (comando non valido)	Il DOS non riconosce il comando. Il comando deve cominciare dalla prima posizione.

32	**SYNTAX ERROR** (ERRORE DI SINTASSI) (comando non valido)	Il comando trasmesso è più lungo dei 58 caratteri consentiti.
33	**SYNTAX ERROR** (ERRORE DI SINTASSI) (nome del file non valido)	La corrispondenza di configurazione è usata in modo non ammesso nei comandi OPEN o SAVE.
34	**SYNTAX ERROR** (ERRORE DI SINTASSI) (nessun file dato)	Il nome del file è stato escluso da un comando o il DOS non lo riconosce come tale. Di solito sono i 2 punti (:) ad essere esclusi dal comando.
39	**SYNTAX ERROR** (ERRORE DI SINTASSI) (comando non valido)	Questo errore può risultare se il comando trasmesso al canale di comando (indirizzo secondario 15) non viene riconosciuto dal DOS.
50	**RECORD NOT PRESENT** (RECORD NON PRESENTE)	Risultato della lettura del disco, al di là dell'ultimo record, attraverso i comandi INPUT o GET. Questo messaggio si presenterà anche dopo il posizionamento su un record oltre la fine del file in un dato file. Se lo scopo è di espandere il file aggiungendo il nuovo record (con un comando PRINT) il messaggio d'errore può essere ignorato. I comandi INPUT o GET non si dovrebbero più tentare dopo il rilevamento di questo errore senza prima effettuare un riposizionamento.
51	**OVERFLOW IN RECORD** (OVERFLOW DEL RECORD)	L'istruzione PRINT supera i limiti del record. L'informazione viene troncata. Dato che il ritorno del carrello, che viene trasmesso come indicatore di fine record, viene calcolato nella dimensione del record, questo messaggio si presenterà se l'insieme dei caratteri del record (incluso il ritorno carrello finale) supera le dimensioni stabilite.
52	**FILE TOO LARGE** (FILE TROPPO GRANDE)	La posizione del record all'interno di un dato file, indica che ci sarà un overflow del disco.

60	**WRITE FILE OPEN** (FILE DI SCRITTURA APERTO)	Questo messaggio viene generato quando un file di scrittura che non è stato chiuso viene aperto per la lettura.
61	**FILE NOT OPEN** (FILE NON APERTO)	Questo messaggio viene generato quando si accede a un file che non è stato aperto nel DOS. A volte, in casi simili, non viene neanche generato un messaggio; la richiesta viene semplicemente ignorata.
62	**FILE NOT FOUND** (FILE NON TROVATO)	Il file richiesto non esiste nel drive indicato.
63	**FILE EXISTS** (FILE ESISTENTE)	Il nome del file che si sta creando già esiste nel dischetto.
64	**FILE TYPE MISMATCH** (CORRISPONDENZA DI TIPO DI FILE ERRATA)	Il tipo di file non corrisponde a quello della posizione sull'elenco per il file richiesto.
65	**NO BLOCK** (NESSUN BLOCCO)	Questo messaggio si presenta in congiunzione col comando B-A ed indica che il blocco da allocare è già stato allocato. I parametri indicano la traccia e il settore disponibili con il numero successivo più alto. Se i parametri corrispondono a zero (0), vengono utilizzati tutti i blocchi di numero più alto.
66	**ILLEGAL TRACK AND SECTOR** (TRACCIA E SETTORE ILLEGALI)	Il DOS ha tentato di accedere a una traccia o a un blocco che non esiste nel formato utilizzato. Ciò può indicare un problema di individuazione del puntatore del blocco successivo.
67	**ILLEGAL SYSTEM T OR S** (TRACCIA O SETTORE ILLEGALE DI SISTEMA)	Questo speciale messaggio d'errore indica una traccia o un settore illegale del sistema.
70	**NO CHANNEL** (NESSUN CANALE) (disponibile)	Il canale richiesto non è disponibile, oppure tutti i canali sono già occupati. Un massimo di cinque file sequenziali può venire aperto contemporaneamente al DOS. I canali ad accesso diretto possono avere sei file aperti.

71	**DIRECTORY ERROR** (ERRORE D'ELENCO)	C'è un problema nell'allocazione BAM oppure sono stati scritti dei dati nella zona riservata alla BAM nella memoria DOS. Per correggere questo problema, reinizializzare il dischetto per ripristinare la BAM nella memoria. Alcuni file attivi possono essere interrotti dall'azione correttiva. NOTA: BAM = Mappa di disponibilità del blocco.
72	**DISK FULL** (DISCO PIENO)	O sono stati usati tutti i blocchi del dischetto oppure è terminato lo spazio disponibile nell'elenco. Il messaggio DISK FULL viene trasmesso quando sul 1541 restano disponibili solo due blocchi per la chiusura del file corrente.
73	**DOS MISMATCH** (73, CBM DOS V2.6 1541)	I DOS 1 e 2 sono compatibili in lettura ma non in scrittura. I dischi possono essere letti indifferentemente con ambedue i DOS, ma un disco formattato in una certa versione non può venire scritto nell'altra versione perché il formato è diverso. Questo messaggio viene visualizzato ogni qual volta venga effettuato un tentativo di scrivere su un disco che è stato formattato in un formato non compatibile (è disponibile una routine di utilità come aiuto alla conversione da un formato all'altro). Questo messaggio può anche apparire dopo l'accensione.
74	**DRIVE NOT READY** (DRIVE NON DISPONIBILE)	È stato fatto un tentativo per accedere all'Unità Floppy Disk senza utilizzare un dischetto.

APPENDICE C

Calcolo di Funzioni Matematiche

Le funzioni che non sono specifiche del BASIC 3.5 possono essere calcolate come segue:

FUNZIONE	EQUIVALENTE BASIC
SECANTE	SEC(X)=1/COS(X)
COSECANTE	CSC(X)=1/SIN(X)
COTANGENTE	COT(X)=1/TAN(X)
SENO INVERTITO	ARCSIN(X)=ATN(X/SQR(-X*X+1)
COSENO INVERTITO	ARCCOS(X)=-ATN(X/SQR (-X*X+1))+ π/2
SECANTE INVERTITA	ARCSEC(X)=ATN(X/SQR(X*X-1)
COSECANTE INVERTITA	ARCCSC(X)=ATN(X/SQR(X*X-1) +(SGN(X)-1*π/2)
COTANGENTE INVERTITA	ARCOT(X)=ATN(X)+π/2
SENO IPERBOLICO	SINH(X)=(EXP(X)-EXP(-X))/2
COSENO IPERBOLICO	COSH(X)=(EXP(X)+EXP(-X))=/2
TANGENTE IPERBOLICA	TANH(X)=EXP(-X)/(EXP(x)+EXP (-X))*2+1
SECANTE IPERBOLICA	SECH(X)=2/(EXP(X)+EXP(-X))
COSECANTE IPERBOLICA	CSCH(X)=2/(EXP(X)-EXP(-X))
COTANGENTE IPERBOLICA	COTH(X)=EXP(-X)/(EXP(X)-EXP(-X))*2+1
SENO IPERBOLICO INVERTITO	ARCSINH(X=LOG(X+SQR(X*X+1))
COSENO IPERBOLICO INVERT.	ARCCOSH(X)=LOG(X+SQR(X)*-1))
TANGENTE IPERBOLICA INV.	ARCTANH(X)=LOG(1+X/(1-X)2
SECANTE IPERBOLICA INVERT.	ARCSECH(X)=LOG(SQR(-X*X+1)+ 1)/X)
COSECANTE IPERBOLICA INV	ARCCSCH(X)=LOG(SGN(X)°SRR(x°x+1)/x)
COTANGENTE IPERBOLICA INV.	ARCCOTH(X)=LOG(X+1)/(x-1)2

APPENDICE D

Tabella delle Note Musicali

NOTA		VALORI SONORI DEI REGISTRI	FREQUENZA EFFETTIVA (HZ)
A	la	7	110
B	si	118	123.5
C	do	169	130.8
D	re	262	146.8
E	mi	345	164.7
F	fa	383	174.5
G	sol	453	195.9
A	la	516	220.2
B	si	571	246.9
C	do	596	261.4
D	re	643	293.6
E	mi	685	330
F	fa	704	349.6
G	sol	739	392.5
A	la	770	440.4
B	si	798	494.9
C	do	810	522.7
D	re	834	588.7
E	mi	854	658
F	fa	864	699
G	sol	881	782.2
A	la	897	880.7
B	si	911	989.9
C	do	917	1045
D	re	929	1177
E	mi	939	1316
F	fa	944	1398
G	sol	953	1575

La tabella precedente mostra i valori sonori dei registri di quattro ottave di note. I valori sonori dei registri sono usati come secondo parametro del comando SOUND. Per utilizzare la prima nota nella tabella (A - valore sonoro dei registri 7) utilizzare il 7 come secondo numero dopo il comando SOUND - SOUND 1,7,30.
Per trovare i valori sonori dei registri per frequenze non comprese nella tabella, utilizzare la formula seguente:

VALORE SONORO DEI REGISTRI = 1024- (111860.781/FREQUENZA)

Sia la tabella dei valori sonori dei registri sia la formula sono applicati a televisori NTSC, il sistema standard utilizzato negli Stati Uniti e in Canada. In paesi che utilizzano il sistema PAL, calcolare i valori sonori dei registri con l'aiuto della formula seguente:
VALORE SONORO DEI REGISTRI = 1024 - (111840.45/FREQUENZA)

APPENDICE E

Codici dei Caratteri di Schermo

Il seguente diagramma elenca tutti i caratteri incorporati nel set di caratteri della Commodore. Mostra i numeri scritti con la funzione POKE nella memoria di schermo (posizioni da 3072 a 4095) per ottenere il carattere desiderato. Viene mostrata inoltre la corrispondenza di un carattere al numero letto dallo schermo con la funzione PEEK.
Sono disponibili due set di caratteri, da utilizzare uno per volta. Questo significa che non è possibile avere sullo schermo i caratteri di un set contemporaneamente ai caratteri dell'altro set. I set vengono selezionati premendo contemporaneamente SHIFT e C=.
Dal BASIC, PRINT CHR$(142) selezionerà la modalità maiuscola/grafica, e PRINT CHR$(14) selezionerà la modalità maiuscola/minuscola.
Tutti i numeri del diagramma possono essere visualizzati INVERTITI. Questo viene ottenuto aggiungendo 128 al valore mostrato.

SET 1	SET 2	POKE	SET 1	SET 2	POKE	SET 1	SET 2	POKE
@		0	T	t	20	(		40
A	a	1	U	u	21	)		41
B	b	2	V	v	22	*		42
C	c	3	W	w	23	+		43
D	d	4	X	x	24	,		44
E	e	5	Y	y	25	–		45
F	f	6	Z	z	26	.		46
G	g	7	[		27	/		47
H	h	8	£		28	0		48
I	i	9	]		29	1		49
J	j	10	↑		30	2		50
K	k	11	←		31	3		51
L	l	12	spazio		32	4		52
M	m	13	!		33	5		53
N	n	14	"		34	6		54
O	o	15	#		35	7		55
P	p	16	$		36	8		56
Q	q	17	%		37	9		57
R	r	18	&		38	:		58
S	s	19	'		39	;		59

SET 1	SET 2	POKE	SET 1	SET 2	POKE	SET 1	SET 2	POKE
<		60		T	84			108
=		61		U	85			109
>		62		V	86			110
?		63		W	87			111
		64		X	88			112
	A	65		Y	89			113
	B	66		Z	90			114
	C	67			91			115
	D	68			92			116
	E	69			93			117
	F	70			94			118
	G	71			95			119
	H	72	spazio		96			120
	I	73			97			121
	J	74			98			122
	K	75			99			123
	L	76			100			124
	M	77			101			125
	N	78			102			126
	O	79			103			127
	P	80			104			
	Q	81			105			
	R	82			106			
	S	83			107			

I codici da 128 a 255 sono le immagini invertite dei codici da 0 a 127.

APPENDICE F

Codici ASCII e CHR$

Questa appendice mostra i caratteri che appariranno battendo PRINT CHR$(X) per tutti i possibili valori di X. Vengono inoltre mostrati i valori ottenuti battendo PRINT ASC("X"), dove X è un qualsiasi carattere battuto. Questo è utile per valutare il carattere ricevuto in un'istruzione GET, per convertire le maiuscole in minuscole, e per stampare i comandi basati sui caratteri (come trasformazione da maiuscolo a minuscolo) che non possono essere tra virgolette.

STAMPA	CHR$	STAMPA	CHR$	STAMPA	CHR$	STAMPA	CHR$
	0	↓	17	"	34	3	51
	1	RVS ON	18	#	35	4	52
	2	CLR HOME	19	$	36	5	53
	3	INST DEL	20	%	37	6	54
	4		21	&	38	7	55
WHT	5		22	'	39	8	56
	6		23	(	40	9	57
	7		24	)	41	:	58
DISABLES SHIFT C=	8		25	*	42	;	59
ENABLES SHIFT C=	9		26	+	43	<	60
	10	ESCAPE	27	,	44	=	61
	11	RED	28	–	45	>	62
	12	→	29	.	46	?	63
RETURN	13	GRN	30	/	47	@	64
SWITCH TO LOWER CASE	14	BLU	31	0	48	A	65
	15	SPAZIO	32	1	49	B	66
	16	!	33	2	50	C	67

STAMPA	CHR$	STAMPA	CHR$	STAMPA	CHR$	STAMPA	CHR$
D	68	♠	97		126		155
E	69		98		127	PUR	156
F	70		99		128	←	157
G	71		100	♠	129	YEL	158
H	72		101	FLASH ON	130	CYN	159
I	73		102	FLASH OFF	131	SPACE	160
J	74		103		132		161
K	75		104	f1	133		162
L	76		105	f3	134		163
M	77		106	f5	135		164
N	78		107	f7	136		165
O	79		108	f2	137		166
P	80		109	f4	138		167
Q	81		110	f6	139		168
R	82		111	HELP	140		169
S	83		112	SHIFT RETURN	141		170
T	84		113	SWITCH TO UPPER CASE	142		171
U	85		114		143		172
V	86	♥	115	BLK	144		173
W	87		116	↓	145		174
X	88		117	RVS OFF	146		175
Y	89		118	CLR HOME	147		176
Z	90		119	INST DEL	148		177
[	91	♣	120		149		178
£	92		121		150		179
]	93	♦	122		151		180
↑	94		123	♣	152		181
←	95		124		153		182
	96		125	♦	154		183

APPENDICE G

Libri per Prodotti Commodore

Le liste seguenti riportano un elenco dei libri disponibili per computer e relativi alla programmazione. Viene riportato per primo il titolo del libro, seguito dall'autore e dall'editore.

Libri Commodore

VIC 20 Programmaer's Reference Guide
Commodore 64 Programmer's Reference Guide
Commodore Plus/4 Programmer's Reference Guide
Mastering Your VIC 20
Four VIC 20 Computer Books:
- VIC Revealed, Nick Hampshire
- VIC Games, Nick Hampshire
- VIC Graphics, Nick Hampshire
- Stimulating Simulations for the VIC, C.W. Engel

Introduction to BASIC, Part 1 and 2, Andrew Colin
Commodore Software Encyclopedia, Third Edition

Programmazione in BASIC

Armchair BASIC: An Absolute Beginner's Guide to
Programming in BASIC, Fox & Fox, Osborne/McGraw-Hill
BASIC Handbook, Second Edition, Lien, Compusoft
Basic Commodore 64 BASIC, Coan, Hayden
Elementary BASIC, Ledgard & Singer, SRA
How to Build a Program, Emmerichs, Dilithium Press
Istant Freeze-Dried Computer Programming in BASIC, Brown
My Computer Likes Me When I Speak in BASIC, Albrecht, Dilithium Press
Nailing Jelly to a Tree, Willis & Danley, Dilithium Press
The Programmer's Book of Rules, Ledin & Ledin, Lifetime
Learning Publishers
Technical BASIC, Kassab, Prentice-Hall

Programmazione in Linguaggio Macchina

Machine Language for Beginners, Mansfield, COMPUTE! Books
Programming the 6502, Zaks, Sybex
6502 Assembly Language Programming, Leventhal, Osborne/Mc Graw-Hill
6502 Micro Chart, Micro Logic Corp
6502 Software Design, Scanlon, Sams
The 6502 Software Gourmet Guide & Cookbook, Findlay, Hayden